



UNIVERSITY OF GENOA
ENGINEERING FACULTY

Ph.D. School “Science and Technology for Information and
Knowledge”

Ph.D. Course “Electronics and Computer Engineering, Robotics
and Telecommunications”

SSD: ING-INF/01

On the Structure of the Hypothesis Space, Model
Selection, and Applications of Statistical Learning
Theory

Sergio Decherchi

Coordinator: Prof. Bruno Bianco

Advisors: Mauro Parodi, Sandro Ridella, Rodolfo Zunino

*With four parameters I can fit an elephant,
and with five I can make him wiggle his trunk.*

John Von Neumann

*I dedicate this work to those who believed in me,
in particular to my Friends, Vera, Carlo and my Parents.*

Acknowledgments

First of all I would like to thank Professors Rodolfo Zunino, Sandro Ridella and Mauro Parodi that built both the Engineer and the Scientist; without their help and contributions all this work would not have been possible; I have been lucky in having three orthogonal instructors.

I would also thank all the past and present members of Sealab: Paolo, Francesco, Giovanni, Judith, Chiara, Fabio, Alessio and Davide. It has been a pleasant experience working with them at Sealab.

Finally I heartily thank my Parents, Carlo and Vera who sustained and believed me during these three years of PhD.

Contents

1	Introduction	3
1.1	Contributions	4
2	Learning and Regularization	7
2.1	Statistical Learning Theory	8
2.1.1	Alternative notions of complexity	13
2.2	Learning as a function approximation problem	18
2.2.1	Reproducing Kernel Hilbert Spaces	20
2.2.2	Kernel Methods	30
2.2.3	Recent Neural Models	42
2.3	Unsupervised Learning, KWM and Random Projections	47
2.3.1	K-Means and Kernel K-Means	47
2.3.2	Plastic Neural Gas	50
2.3.3	Spectral Clustering	50
2.3.4	Random Projection	53
2.3.5	KWM Classifiers and Prediction Error Estimation	54
2.4	Model Selection	56
2.4.1	Test Set method	56
2.4.2	K-fold cross validation	56
2.4.3	Leave-One-Out	57
2.4.4	Bootstrap	57
2.4.5	Generalization error bounds	57
3	Structuring the hypothesis space	59
3.1	The Regularized Mean Problem	60
3.1.1	Links with James-Stein theory	62
3.1.2	Fundamental Laws of the Regularized Mean Problem	64
3.1.3	A Machine Learning Approach to the Mean Value Estimation	68
3.1.4	Conclusions	82
3.2	Generalized Tikhonov	85

CONTENTS	V
3.2.1 Generalized Tikhonov and Oracular Regularization . . .	86
3.2.2 On Learning	90
3.2.3 On Shrinking	93
3.2.4 On Filtering	99
3.2.5 Non linear extensions	103
3.3 Discussion and Conclusion	105
3.4 Biased Regularization for Controlling Capacity	107
3.4.1 Constraining SVM Capacity by Unsupervised Analysis .	108
3.4.2 Algorithm for Constrained Optimization	115
3.4.3 Experimental Results	119
3.4.4 Conclusions	128
3.5 Semi-Supervised Learning by Biased Regularization	131
3.5.1 Biased Regularization	132
3.5.2 Biased SVM	135
3.5.3 Biased RLS	137
3.5.4 Semi-Supervised Learning by using Biased Regularization	138
3.5.5 Experimental Results	144
3.5.6 Conclusions	151
3.6 Explicit Transductive bound	153
3.6.1 Induction, Transduction and Semi-Supervised learning	153
3.6.2 Overall Risk Minimization	154
3.6.3 Bound derivation	156
3.6.4 Valuation and experimental results	158
4 Applications	160
4.1 Text Clustering for Security Applications	161
4.1.1 Document Clustering in Text Mining and Security . . .	162
4.1.2 Hybrid Approach to Kernel K-Means Clustering	168
4.1.3 The document-clustering framework	172
4.1.4 Experimental Results	175
4.1.5 Conclusions	185
4.2 SVM Analog Circuit Based Learning	187
4.2.1 Hardware SVM	187

CONTENTS	VI
4.2.2 Co-Content Minimization Circuits	188
4.2.3 Hardware SVM Training	189
4.2.4 Experimental Results	194
4.2.5 Conclusions	196
4.3 Fast Approximate Regularized Least Squares	201
4.3.1 Toeplitz Matrixes for Regularized Least Squares	202
4.3.2 Experimental Results	209
4.3.3 Concluding Remarks	216
4.4 Regularized Random Neural Networks	217
4.4.1 Generalization ability of the basic ELM model	217
4.4.2 Augmenting ELM with a Regularization Term	219
4.4.3 Regularization Strategies for ELM	220
4.4.4 Experimental Results	223
4.4.5 Conclusions	225
4.5 Efficient Covariate Shift Detection by Clustering	228
4.5.1 Non Stationarity Detection for Assessing the applicabil- ity of generalization error estimation	229
4.5.2 Discussion	233
4.5.3 Experimental Results	236
4.5.4 The Email Spam Database	247
4.5.5 Conclusions	250
4.6 Underwater Port Protection by Machine Learning Tools	252
4.6.1 Architectures for Magnetic-based Detection Systems	253
4.6.2 Machine Learning Methods for Magnetic-based Detec- tion	256
4.6.3 Overall Architecture of the Adaptive Detection System	257
4.6.4 Experimental Results	259
4.6.5 Conclusions	267
5 Conclusions	269
Appendices	273

CONTENTS	VII
A Regularized Mean Problem	273
A.1 Proof of 3.1.1	273
A.2 Proof of the variance, 3.1.2	273
A.3 Proof of degrees of freedom, 3.1.1	274
A.4 Proof of 3.1.3	276
A.5 Proof of 3.1.4	277
A.6 Proof of Leave One Out 3.1.2	278
A.7 Proof of 3.1.5	279
A.8 Proof of Maximal Evidence 3.1.3	280
B VQSVM section	282
B.1 Proof of Theorem 3.4.2.1:	282
B.2 Proof of Lemma 3.4.1	283
B.3 Proof of Lemma 3.4.2	283
B.4 Proof of Lemma 3.4.3	284
C Biased Regularization section	285
C.1 Proof of Biased SVM dual theorem	285
C.2 Proof of Biased RLS primal solution	286
C.3 Proof of Biased RLS dual	287
D Generalized Tikhonov section	289
D.1 Proof of Theorem 3.2.1	289
D.2 Proof of 3.64	290
D.3 Proof of kernel matrix eq.(3.71)	292

List of Figures

2.1	Loss functions for classification: blu dotted line is hinge loss, red '+' line is logistic loss, black dashed line is square loss	28
2.2	Circular Back Propagation Network with the circular input evidenced	43
3.1	Advantage of oracular regularized solutions against non regularized: x axis is μ range, y axis is the quality metric	62
3.2	Multiple Kernel Learning average kernel weights values for $\mu \in [0, +1]$: x axis is each feature (kernel) and y axis is the average associated weight	72
3.3	Multiple Kernel Learning average kernel weights values for $\mu \in [-1, 0]$ x axis is each feature (kernel) and y axis is the average associated weight	73
3.4	Dashed line is w_{MAX} and w_{MIN} is continuous line: x axis is μ value and y axis are w_{MAX} and w_{MIN}	75
3.5	x axis is μ value and y axis is the quality metric (3.32). Continuous line stands for $\hat{\lambda}^{orac}$, and dashed line for λ_{ht}^{loo}	77
3.6	Point-Dashed line for λ_{β}^{orac} , continuous line is for λ_{β}^{loo} and dashed line is for $\hat{\lambda}_{\beta}^{orac}$: x axis is μ value and y axis is the quality metric (3.32)	78
3.7	Behavior of the quality metric y when using λ_{β}^{loo} : x axis is μ value with varying sigma in the range $[0.2, 2]$ (step 0.2). Highest peak curve corresponds to the highest value of the parameter σ	78
3.8	Neural scheme	80
3.9	Neural Regularizers results: x axis is μ value and y axis is the quality metric	81
3.10	Neural Regularizers results on the domain $\mu_M = 1$: x axis is μ value and y axis is the quality metric	83
3.11	Neural Regularizers results: x axis is μ value and y axis is the percentual gain for $m = 50$	84
3.12	Agnostic Oracular Vectors for the univariate case and $w_* = 1$	91

LIST OF FIGURES**IX**

3.13 Regularizers for the d -means problem	97
3.14 Wiener vs Robust filtering	104
3.15 Wiener vs Robust filtering 2D	105
3.16 Tikhonov equivalence scheme and generalized learning,shrinking and filtering	106
3.17 Relative positions of the solution vector, w , with respect to the unsupervised reference, $w^{(KM)}$, from left: Case 1): Within the hypersphere; Case 2): on the hypersurface; Case 3): Out of the hypersphere	115
3.18 Model selection surfaces in the hyper-parameter space: a) Conventional- SVM MD-bound surface . b) Constrained-SVM MD bound sur- face c) Validation-error surface	123
3.19 Comparison of conventional and constrained SVM model se- lection methods. a) Generalization bounds b) Validation error	123
3.20 Role of λ_2 in biased regularization	134
3.21 Inductive Bias and Hypothesis space	142
3.22 Two moons semi supervised learning	145
3.23 Semi-supervised text mining	147
3.24 Semi-supervised MD text mining	148
3.25 USPS Comparison with different reference clusterings	149
3.26 USPS Comparison with LapRLS, LapSVM, TSVM	150
3.27 Isolet Comparison with LapRLS, LapSVM, TSVM	151
3.28 Experiment for $k = l$ variable. Note the advantage of transduc- tive bound (red line) over induction (blue line)	159
4.1 A generic process model for a document-clustering application	164
4.2 A document partitioned in 3 sections and terms. $T = t_j; j = 1, \dots, n_T$ Gaussian densities in each section	172
4.3 A document partitioned in 3 sections, terms $T = t_j; j = 1, \dots, n_T$ and vector v'' representation Gaussian densities in each section	172
4.4 The cumulative distributions of the cluster purity for the ex- periments reported in Table 4.1	179
4.5 Results of the second experiment on the Enron database	185

LIST OF FIGURES**X**

4.6	Multi-terminal resistive network connected to capacitors . . .	188
4.7	Circuit topology: three terminal case	192
4.8	Voltage limiting component and voltages-current curve	192
4.9	Complete circuit	192
4.10	Sonar dataset: dashed line represents the software accuracy, continuous line is the circuital complexity and dashed-dotted line is svm hardware accuracy	195
4.11	Ionosphere dataset: dashed line represents the software accu- racy, continuous line is the circuital complexity and dashed- dotted line is svm hardware accuracy	195
4.12	Diabetes dataset: dashed line represents the software accu- racy, continuous line is the circuital complexity and dashed- dotted line is svm hardware accuracy	195
4.13	Kernel space data representation (crosses are +1 data and balls are -1 data): (a) is the case $\sigma \rightarrow 0$, (b) is the case $\sigma \rightarrow \infty$	209
4.14	Normalized Frobenius norm of the error matrix versus kernel amplitude.	213
4.15	Accuracy comparison for Gaussian Elimination, Conjugate Gra- dient and Toeplitz Approximation	213
4.16	Experimental results for the RLS Toeplitz-based acceleration.	214
4.17	Classification error for ELM and Regularized ELM	225
4.18	RMSE for Regression problems using ELM and Regularized ELM	226
4.19	2-D artificial dataset with a non-stationary data distribution. a) Training Set, X_{tg} b) Stationary Test Set, X_{ts} c) Test set, X_{ts1} d) Test set, X_{ts2} . e) Test set, X_{ts3}	237
4.20	Discrepancy measurements for the 2-D artificial experiment. The curves are parameterized by the number of prototypes, n_h . Discrepancy values increase in the presence of non-stationary distributions of data.	240
4.21	Generalization bounds and true classification performances. Theoretical predictions may prove unreliable by the presence of non-stationary distributions; X_{ts} is the only case involving a stationary distribution.	240

4.22	
(a)		
243subfigure.4.22.1		
4.23	MNIST OCR domain: Predicted error performances and actual generalization performances for stationary and non-stationary test sets	246
4.24	Comparison between the adaptive (A) and the classical (B) detection approach on a real diver-intrusion test. The classical filtering approach (involving the total magnetic field only) induces a false-positive.	254
4.25	RIMAN adaptive configuration: the array of magnetometers are referred to one reference sensor (MAG0). Each blocks ΔF_{x0} supports the adaptive target detection of the x -th sensor.	255
4.26	SIMAN configuration: each magnetometer operates both as a sentinel and as a reference for neighboring devices. Block ΔF_{xy} supports the adaptive target detection for the pair of sensors MAG _x and MAG _y	255
4.27	The processing architecture of the overall neural magnetic-based detection system.	259
4.28	Signal-classifier data analysis: 2-dim Random Projections for increasing window overlap, L	261
4.29	Event-classifier data analysis: 2-dim Random Projections for increasing window overlap, L	261
4.30	Unsupervised data analysis: optimal number of clusters vs increasing window overlap, L (dashed line indicates Signal-classifier data, the solid line indicates Event-classifier data)	262
4.31	PGAS classification performances; a) - Signal classifier; b) - Event classifier; c) - Overall System performance	263
4.32	Signal classifier: measured test errors for different kernel parameters	263
4.33	Event classifier: measured test errors for different kernel parameters	264
4.34	Signal Classifier: test errors for optimal kernel par.	264

LIST OF FIGURES	XII
4.35 Event Classifier: test errors for optimal kernel par.	264
4.36 Global classification accuracy for optimal kernel par. and vary- ing window size	264
4.37 CBP, Signal/Event Accuracies	265
4.38 Overall detection performances when using CBP networks . .	265
4.39 Measured false-positive rates for SVM classifiers	267

List of Tables

3.1	Regularization, Learning, Filtering, and Shrinking functional form	106
3.2	MNist digit recognition. Model-Selection results, bounds, and accuracy of conventional svm and constrained svm.	125
3.3	Newsgroup-20 binary problems	126
3.4	Newsgroup-20 results	126
3.5	“Heart” testbed. Comparison between classical MD and VQSVM generalization bounds	128
3.6	“Heart” testbed. Measured test error for model selection validation	128
3.7	“Ionosphere” testbed. Comparison between classical MD and VQSVM generalization bounds	128
3.8	“Ionosphere” testbed. Measured test error for model selection validation	129
3.9	“Sonar” testbed. Comparison between classical MD and VQSVM generalization bounds	129
3.10	“Diabetes” testbed. Comparison between classical MD and VQSVM generalization bounds	129
3.11	“Diabetes” testbed. Measured test error for model selection validation	129
3.12	Comparison among semi supervised classificatio methods . .	144
4.1	Clustering performances obtained on Reuters-21578 with $\alpha = 0.3$	178
4.2	Clustering performances obtained on Reuters-21578 with $\alpha = 0.5$	178
4.3	Clustering performances obtained on Reuters-21578 with $\alpha = 0.7$	179
4.4	Clustering performances obtained on Reuters-21578 with $\alpha = 0.3$ and dimension reduction $n_r = 500$	180
4.5	Clustering performances obtained on Reuters-21578 with $\alpha = 0.3$ and dimension reduction $n_r = 100$	181
4.6	Clustering performances obtained on \mathcal{D}_{N_1} with $\alpha = 0.3$	182
4.7	Clustering performances obtained on \mathcal{D}_{N_2} with $\alpha = 0.3$	182

4.8	Results of the first experiment on the Enron dataset: j is cluster index and $ C_j $ is the cluster size	183
4.9	Datasets Splitting and Ω parameters	195
4.10	Comparison of SW SVM and HW SVM accuracy for Sonar Dataset	197
4.11	Comparison of SW SVM and HW SVM accuracy for Ionosphere Dataset	198
4.12	Comparison of SW SVM and HW SVM accuracy for Diabetes Dataset	199
4.13	Comparison among Gaussian Elimination, Conjugate Gradient and the Toeplitz-based solver approaches to RLS learning: n is the number of patterns and k is the number of CG iterations	208
4.14	Data splitting criteria for the data sets used in the experiments and number of variables. The numbers of patterns in the table are intended multiplied by $1e3$	210
4.15	Theoretical formulation of divergence measures derived from the general class of f-divergences	232
4.16	Kullback-Liebler divergence D_{KL}	238
4.17	Hellinger divergence D_H	238
4.18	Total Variation D_{TV}	238
4.19	Pearson (Chi-Square) D_P	239
4.20	Training error R_{emp} , bound, and actual classifications error for the artificial dataset pairs	239
4.21	KDD99: measured divergence values for the original distribution (T, S) and the stationary reference (T_J, S_J)	242
4.22	KDD99: training error R_{emp} , actual error R , bound, for (T, S) and the stationary reference (T_J, S_J)	243
4.23	MNIST OCR domain, D_{KL} values	245
4.24	MNIST OCR domain, D_H values	245
4.25	MNIST OCR domain, D_{TV} values	245
4.26	MNIST OCR domain, D_P values	246
4.27	MNIST OCR predicted and empirical classification errors for the stationary pair (X_{tg}, X_{ts}) and the validation pair (X_{tg}, X_{val}) .	246
4.28	Spam Assassin: divergence values for (T, S) and (T_J, S_J)	248

LIST OF TABLES**XV**

4.29 Spam Assassin: training classification error R_{emp} , actual error R , bound	248
4.30 Daimler: divergence values for (T, S) and (T_J, S_J)	249
4.31 Daimler: training classification error R_{emp} , actual error R , bound	249
4.32 Number of patterns for the Signal Classifier	260
4.33 Number of patterns for the Event Classifier	260

Abstract

Learning Theory, and in particular Statistical Learning Theory, is the theory that mathematically formalizes the conditions and the algorithms that allow to define (*learn*) empirical models from a given, finite number of samples.

A major open point in Statistical Learning Theory is the definition of a theoretically justified method to select a model that properly describes data for a specific task (e.g. classification, regression, clustering, ranking, collaborative filtering). This problem is tightly connected to that of the definition of a proper hypothesis space, i.e., a proper class of functions that allow learning. This dissertation inquires on the problem of giving a structure to the hypothesis space by performing different analysis.

A first analysis shows how structuring the hypothesis space of Tikhonov regularization allows to build bridges with Wiener filtering and Shrinking. In particular, the definition of a closed form solution is provided to obtain oracular (i.e. optimal) regularization. Furthermore, from another perspective, it is shown that in the oracular case the James-Stein paradox does not take place. In a second analysis, biased Regularization is used to order and shrink the space of functions available at learning time. The ordering/reduction is obtained by using a reference function that approximately describe the learning goal. The key observation is that any reasonable reference function improves on the non ordered, usual hypothesis space given by the identically null function . It is observed that in a wide class of problems, where a clustering hypothesis exists, the reference function can be given by a clustering process. A natural consequence of this approach is that, by biased regularization, one gets a new class of Semi-Supervised Learning algorithms that merges the clustering results with potentially any kernel machine. The ultimate outcome is that only a minimal modification to the baseline learning algorithms is needed; thus, efficient implementations are possible, and their effectiveness is proved by experimental validation. These learning methods inherit the intrinsic property of shrinking the generalization error bounds, thus allowing effective model selection with very few labeled data. Finally, several

applications and extensions of existing learning methods are presented.

1

Introduction

Machine Learning [1] is a fascinating interdisciplinary topic where algebra, geometry, statistics, probability theory, and to some extent cryptography [2] merge together. The purpose of Machine Learning techniques is to infer properties on unseen data given a previous *learning* or *training* stage, during which the relationship f between the data and the properties to be inferred is derived. In other words, machine learning supports systems that can abstract knowledge from data rather than simply memorize a set of rules for labeling them.

The word *learning* stands for an algorithmic procedure by which, from a finite number of examples, the inductive rule f is obtained. It is vital in a learning process that the rule f endows a certain abstraction level: abstraction means that f is able to effectively predict on unseen samples, thus it *generalizes*.

Every intelligent system, intuitively, should memorize *something* but should not need to memorize everything. In this respect, the interpolation of some given points can be seen as a degenerate case of learning. In the interpolation process, one is constrained to perfectly fit every datum: no freedom is allowed and the interpolation function f is hardly constrained. In this case, a memorization process is carried out, rather than an abstraction one; as soon as the hard constraint *f must fit every data* is relaxed, then one is moving from memorization towards learning. From another point of view, learning, and thus abstraction, implies compression of a representation due to the skill

of finding redundancies on given data. Cryptography is exactly the opposite of learning as its purpose is to make data non-understandable (i.e. with no redundancies, and so non compressible). A good cypher is everything that cannot be learned from examples; within this view learning is related to code-breaking with a polynomial complexity algorithm.

Learning and cryptography are the two opposite arrows of information theory: this thesis concentrates only on the first direction, that is learning in particular the Statistical Learning Theory of Vapnik and Chervonenkis [3] and kernel methods is considered as starting point.

This thesis, from the theoretical point of view, analyzes the problem of structuring the hypothesis space: by this expression one means that the usual regularization strategy is somehow *flat* in selecting the class of functions used to learn: this is in contraposition with an aggressive regularization strategy where functions are *ordered* by some criterion: in particular notions of *oracular* or sample dependent *priors* suitable for kernel methods will be proposed and studied in order to *rank* the hypothesis space.

From the applicative perspective, the thesis proposes several engineering applications and algorithmic findings.

The thesis is organised in three parts: the first introduces the basic concepts (Chap.2) used throughout the thesis; third and fourth chapters are original contributions. Chapter 5 gives some conclusions.

1.1 Contributions

The contributions of this thesis can be summarized in the following points: theoretical contributions regarding the structure of the hypothesis space (Chap.3), and various applicative and algorithmic findings (Chap.4). The first part contributes with:

- The regularized mean problem, introduced in [4],[5], is the simplest form of Tikhonov regularization. This preliminary work on the mean estimation showed the limits of regularization in this context according to James-Stein theory. Neural networks are proposed as viable tools to

compute the mean in noisy and small sample cases. It is shown, that given a proper prior knowledge, neural networks can predict the mean value better, in expectation, with respect to the sample mean formula. Moreover a notion of *oracular regularization* is introduced: here *oracular* means that no better solution can be achieved and that the regularization strategy uses information about the optimal solution itself.

- A further study [6] extends the analysis of the mean to Tikhonov regularization. A justified notion of oracular regularization consistent with that of the mean is proposed; this leads to the consequence that for an *optimal learning*, in expectation, a generalized Tikhonov functional must be used in which an oracular regularization operator can be defined. This work shows that the Vapnik concept of Universum [7] is useful to get an optimal regularization strategy, moreover an ideal Universum can be defined that reflects oracular regularization. The developed study suggests that a generalization of the regularization operator can be significant. Further this work poses a bridge between learning, regularization, filtering and shrinking; indeed all these problems can be dealt through the same formalism. For instance, it is shown that Wiener filter is exactly an instance of oracular Tikhonov regularization, at the meantime in the shrinking context it is shown that using oracular regularization the James-Stein paradox disappears.
- A study on biased regularization: when using biased regularization one gives a prior on the norm of a Reproducing Kernel Hilbert Space \mathcal{H} and instead of using the usual one a biased version is employed. This model has been developed [8] first for SVM using a Ivanov like regularization term [9] with the aim of reducing the space of functions and thus showing that data dependent generalization error bounds shrink accordingly. Then the Tikhonov-like version of biased regularization has been used to derive bSVM, bRLS [10] are the biased versions of SVM, RLS. Such machines can be used to perform Semi-Supervised Learning by a slight modification of the original algorithms; moreover the property that regards the bound shrinkage is preserved thus allowing effective model

selection with very few labeled data. In particular for bSVM it is shown that an efficient learning algorithm can be given. The technique essentially performs clustering on data by any clustering algorithm, maps the learned partition by the base version of the kernel machine in use into a reference function, and then learns via the biased machine on labeled data where biasing is given by the reference function. This method proved fast and when coupled with spectral clustering gives results at the state of the art. A software is freely available at http://www.sealab.dibe.unige.it/biased_learning that compares the proposed approach to TSVM, LapRLS and LapSVM.

- An explicit bound for Transductive learning based on Vapnik Chervonkis dimension [11].

The second part contributes with (Chap. 4):

- An analogic circuit implementation of SVM learning stage [12],[13]
- A fast approximated algorithm for RLS learning based on properties of Toeplitz matrix [14], suitable for DSP or low-power, low-cost devices.
- A simple method to assess by clustering the non stationarity of two given datasets [15],[16]
- The development of a clustering/classification engine for text mining [17],[18],[19],[20],[21],[22]
- A study on regularized random neural networks [23]
- Application of SVM, Plastic Neural Gas, and Circular Back propagation for divers detection [24],[25] in underwater port protection.

2

Learning and Regularization

Aim of this chapter is to introduce Statistical Learning Theory and the algorithmic tools on which the rest of the thesis is based on. When dealing with learning theory there are essentially four kind of problems: supervised learning, unsupervised learning, semi-supervised learning, and transductive problems. In the first class one is given n couples (\mathbf{x}, y) where $\mathbf{x} \in \mathbf{R}^d$, y is real and the task is finding a function f that given every \mathbf{x} is able to predict the corresponding y . In unsupervised learning one is given only the vectors \mathbf{x} and labels are not present: the goal here is to perform unsupervised operations such as for instance clustering. In the semi-supervised learning task one still wants to find an f able to predict y , but this time, training data is both labeled and unlabeled. Finally in the transductive task one is given both labeled and unlabeled data; now the task is predicting only on the unlabeled data and not inferring a general f usable for any \mathbf{x} .

Statistical learning theory (SLT) is the theory that allows to understand what are the conditions and the methods by which effective learning of f can happen: SLT deals with supervised and transductive learning.

2.1 Statistical Learning Theory

Statistical Learning Theory (SLT) [3] is by far one of the main milestones of recent machine learning theories: SLT explains what are the conditions by which learning can take place. SLT tries to explain what in classical statistics is called the *bias variance dilemma*; in the terminology of learning this means finding a model, f , of the data, e.g. regression problems, that both is not *over-smoothing* and not *overfitting*: *over-smoothing* means that the model f is too simple with respect to given data, that is, f tends to ignore data (abstraction); *overfitting* conversely means that f fits too much the given data, f tends to memorize data.

SLT has the merit of having developed tools able to statistically formalize the problem of finding an equilibrium between abstraction and memory in learning problems. The operation of finding the equilibrium itself is an important problem and is called model selection: SLT does not solve this problem but is able to say what are the fundamental ingredients that are needed to solve the problem.

SLT makes the assumption that, whatever is the task, data are sampled from an unknown probability distribution $\mathcal{P}(\mathbf{x}, y)$ where \mathbf{x} are samples $\in \mathcal{X} \cap \mathbb{R}^d$ and $y \in \mathcal{Y} \cap \mathbb{R}$ in regression problems or $y \in \{+1, -1\}$ in classification problems. The learning problem is to minimize the *risk*:

$$R[f] = \int_{\mathcal{X} \times \mathcal{Y}} L(\mathbf{x}, y, f(\mathbf{x})) d\mathcal{P}(\mathbf{x}, y) \quad (2.1)$$

where L is a loss function, that is an error measure as for instance $(y_i - f(\mathbf{x}_i))^2$. The difficulty of the learning process is given by the fact that $\mathcal{P}(\mathbf{x}, y)$ is unknown and one only has a finite set of couples (\mathbf{x}_i, y_i) . One way to approximate (2.1) given n patterns is to minimize the *empirical risk*:

$$R_{emp}[f] = \frac{1}{n} \sum_{i=1}^n L(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) \quad (2.2)$$

The principle by which one minimizes $R_{emp}[f]$ is called *Empirical Risk Minimization* ERM. A central observation consists in noting that $f \in \mathcal{F}$ and \mathcal{F} is a rich enough function space, than one can always get $R_{emp}[f]$ almost 0. From

now on, one concentrates on classification problems: for instance consider using the following function:

$$\begin{cases} f(\mathbf{x}) = 1 & \text{if } \mathbf{x} = \mathbf{x}_i \text{ for some } i = 1, \dots, k \\ f(\mathbf{x}) = -1 & \text{otherwise} \end{cases} \quad (2.3)$$

this function clearly leads to $R_{emp}[f] = 0$. However for a new point \mathbf{x}_j the prediction will always be +1 class thus nothing has been learned from the given data. The key concept is that if no restrictions are applied on \mathcal{F} than learning cannot take place; thus one needs a way to take into account the size or the *capacity* of a function space \mathcal{F} . From this capacity control coupled with adherence to the given data than one can hope to learn. From the Bayesian perspective this amounts to a prior of the functions f .

Using the tool of Chernof bound [3] one can show that for a given f it holds:

$$P\{|R_{emp}[f] - R[f]| \geq \epsilon\} \leq 2e^{-2n\epsilon^2} \quad (2.4)$$

This result at a first sight seems extremely good: it says that for a given f the $R_{emp}[f]$ approaches $R[f]$ at exponential speed with respect to n . However, this bound gives information only about a specific f , conversely, a learning system deals with an entire class of functions \mathcal{F} and not only one f ; from the math point of view if f is learned than in $R[f]$ data and f are not independent and thus the previous bound is not valid. Another further requirement is *consistency*: *consistency* amounts to request that the the minimum $R_{emp}[f]$ asymptotically $n \rightarrow \infty$ is the same as that of $R[f]$. It can be shown [3] that ERM is not consistent without giving any restriction on \mathcal{F} ; Vapnik-Chervonenkis theory shows that the *worst case* of the functions in \mathcal{F} is the key to grant consistency of ERM. The request of consistency can be translated in a request for uniform convergence in probability [3]; in bounds terms:

Theorem 2.1.1. *One-sided uniform convergence in probability:*

$$\lim_{n \rightarrow \infty} P\{\sup_{f \in \mathcal{F}} (R[f] - R_{emp}[f]) > \epsilon\} = 0 \quad (2.5)$$

for all ϵ , is a necessary and sufficient condition for non trivial consistency of empirical risk minimization.

In the case of two-sided convergence one refers to Glivenko-Cantelli classes of functions f . There are two possibilities now on the space of functions \mathcal{F} . The first is that $|\mathcal{F}|$ is finite and the second is that this cardinality is not finite. Suppose now that the number of possible models is finite and its value is $|\mathcal{F}|$; applying by Bonferroni correction [3] it can be shown that the following bound holds for every function f :

$$P\{|R[f] - R_{emp}[f]| > \epsilon\} < |\mathcal{F}|e^{-2\epsilon^2 n} \quad (2.6)$$

Denoting by $\delta = 2|\mathcal{F}|e^{-2\epsilon^2 n}$ and solving for ϵ one can rewrite this result as:

$$R[f] \leq R_{emp}[f] + \sqrt{\frac{\ln |\mathcal{F}| + \ln \frac{2}{\delta}}{2n}} \quad (2.7)$$

that holds with confidence $1 - \delta$.

This result is fundamental because clearly shows the two ingredients of learning: the first is the empirical risk that is the adherence to the data, the second takes into account the cardinality of the space of the models $|\mathcal{F}|$. One can go further and use the fact that:

$$\sqrt{\frac{\ln |\mathcal{F}| + \ln \frac{2}{\delta}}{2n}} \leq \sqrt{\frac{\ln |\mathcal{F}|}{2n}} + \sqrt{\frac{\ln \frac{2}{\delta}}{2n}} \quad (2.8)$$

to show that:

$$R[f] \leq R_{emp}[f] + \sqrt{\frac{\ln |\mathcal{F}|}{2n}} + \sqrt{\frac{\ln \frac{2}{\delta}}{2n}} \quad (2.9)$$

This further specification shows that, to be precise, the terms to get into accounts are three: the first is the empirical risk, the second is the cardinality of space of functions and the third is about the finiteness of sample: this last term is purely computable by the confidence on the bound $1 - \delta$ and the number of samples.

These facts are true when the number of models is finite: however often it can happen in real classifiers that the cardinality of space of functions is not finite; thus Bonferroni correction cannot be used to build a valid bound; other tools and capacity concepts are needed.

Let $X = \mathbf{x}_1, \dots, \mathbf{x}_n$, as usual, be a random independent observation of size n . Let

$$N^{\mathcal{F}}(X) \leq 2^n \quad (2.10)$$

be the number of possible separations of the sample X by a given set of functions \mathcal{F} ; this quantity is usually called the *shattering coefficient* (its corresponding concept on regression is the *covering number*). Assume that $N^{\mathcal{F}}(X)$ is measurable with respect to a probability measure $p(X)$; for this reason the expectation of $N^{\mathcal{F}}(X)$, $E_X[N^{\mathcal{F}}(X)]$ is a well defined quantity. A key quantity that can be defined is the *Annealed Entropy*:

$$H_{ann}^{\mathcal{F}}(n) = \ln E_X[N^{\mathcal{F}}(X)] \quad (2.11)$$

In particular it can be shown that $H_{ann}^{\mathcal{F}}(n)$ is an appropriate concept of *complexity* of the function space \mathcal{F} . In particular the following milestone result holds:

Theorem 2.1.2. *For the existence of non trivial exponential bounds on uniform convergence, the relation*

$$\lim_{n \rightarrow \infty} \frac{H_{ann}^{\mathcal{F}}(n)}{n} = 0 \quad (2.12)$$

is a sufficient condition. In particular the following bound holds true for any $f \in \mathcal{F}$ with probability $1 - \delta$:

$$R[f] \leq R_{emp}[f] + \frac{\chi(n)}{2} \left(1 + \sqrt{1 + \frac{4R_{emp}[f]}{\chi(n)}} \right) \quad (2.13)$$

where:

$$\chi(n) = \epsilon^2 = 4 \frac{H_{ann}^{\mathcal{F}}(2n) - \ln \delta/4}{n} \quad (2.14)$$

This bound is equivalent to its corresponding version for $|\mathcal{F}|$ finite, where $H_{ann}^{\mathcal{F}}(2n)$ has the role of \mathcal{F} . One could expect in the bound the quantity $H_{ann}^{\mathcal{F}}(n)$ instead one gets $H_{ann}^{\mathcal{F}}(2n)$; this issue is due to the rather technical concept of symmetrization, that is a technique used in the proof.

This last result formula, is conceptually elegant, but is simply not computable in real scenarios; for this reason one has to build surrogate concepts of complexity till one gets something really computable. One first defines the concept of *Growth function*:

$$G^{\mathcal{F}}(n) = \sup_X \ln N^{\mathcal{F}}(X) \quad (2.15)$$

Since the *Growth function* does not depend on the probability measure and is not less than the annealed entropy, then:

$$H_{ann}^{\mathcal{F}}(n) \leq G^{\mathcal{F}}(n) \quad (2.16)$$

By analyzing the structure of Growth function and using the Sauer lemma [3] one can obtain the further following bound:

$$G^{\mathcal{F}}(n) < d_{vc} \left(1 + \ln \frac{n}{d_{vc}} \right) \quad (2.17)$$

where d_{vc} is a central quantity and is called Vapnik-Chervonenkis dimension. This quantity can be defined in the following way:

Definition 2.1.1. *The VC dimension of a set of functions \mathcal{F} is equal to the largest number d_{vc} of vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$ that can be separated into two different classes in all the $2^{d_{vc}}$ possible ways using this set of functions.*

One could note that if for any n exists a set of n vectors that can be *shattered* by the functions \mathcal{F} then $d_{vc} = \infty$. Thus one can employ this notion of complexity in the generalization error bound and getting computable quantities.

The d_{vc} is a very important and critical concept; d_{vc} is not necessarily equal to the number of free parameters of a learning system. For instance it can happen that d_{vc} of set of non linear functions can exceed the number of parameters; let consider this case with an example.

Consider the following set of functions parametrized by $w \in (0, \infty)$:

$$f(x, w) = \text{sign}[\sin(\pi wx)]. \quad (2.18)$$

where $x \in (0, 2\pi)$. The thesis is that by only w one can get $d_{vc} = \infty$. This is equivalently rephrased as establishing that for any n and for any binary sequence:

$$\delta_1, \dots, \delta_n \quad \delta_i \in \{0, 1\} \quad (2.19)$$

there exist n points such that the system of equations

$$\text{sign}[\sin(wx_i)] = \delta_i \quad (2.20)$$

has a solution in w . Consider the points $x_i = 2\pi 10^{-i}$, $i = 1, \dots, n$. One can verify that for these points the value:

$$w = 0.5 \left(\sum_{i=1}^n (1 - \delta_i) 10^i + 1 \right) \quad (2.21)$$

gives a solution to the system of equations (2.20). For this reason the number of parameters does not determine the d_{vc} ; rather it is the opposite.

2.1.1 Alternative notions of complexity

The exposed theory derives different notions of complexities, namely, Shattering coefficient, Annealed Entropy, Growth function and d_{vc} : during the process of continuous bounding one has the practical problem that d_{vc} estimates give generalization bounds extremely loose ($R[f] < 120\%$ is not so useful). Moreover d_{vc} derives from a worst-case analysis a most of the information about data is lost. Despite Vapnik bounds well explain the machinery behind the process of learning, from the practical point of view other bounds are needed to assess the generalization error. Bounding techniques developed for this issue are: Maximal Discrepancy [26], PAC Bayesian bounds [27], Rademacher complexity [28] and Stability bounds [29]: the first three will be briefly discussed due to their real world effectiveness.

2.1.1.1 Maximal Discrepancy (MD), and Rademacher complexity (RC)

These two notions of complexity are tightly linked and are based on the same intuitive idea: *if your class of functions \mathcal{F} is able to fit noise this may mean that your class is too big and you will probably overfit*. One should note that d_{vc} is so conservative that deals with the worst possible random noise; somehow MD and RC deal with a reasonable random noise.

The definition of maximal discrepancy is the following [28]:

Definition 2.1.2. *Let $p(X)$ be a probability distribution on a set X and suppose that $\mathbf{x}_1, \dots, \mathbf{x}_n$ are independent samples selected according to $p(X)$. Let \mathcal{F} be a class of functions mapping from X to \mathbb{R} . The maximal discrepancy of \mathcal{F} is the*

random variable:

$$\mathcal{D}_n(\mathcal{F}) = \sup_{f \in \mathcal{F}} \left(\frac{2}{n} \sum_{i=1}^{n/2} f(\mathbf{x}_i) - \frac{2}{n} \sum_{i=n/2+1}^n f(\mathbf{x}_i) \right) \quad (2.22)$$

Another key quantity, is the Empirical Rademacher Complexity; this is the following random variable:

$$\hat{\mathcal{R}}_n(\mathcal{F}) = \mathbf{E}_\sigma \left(\sup_{f \in \mathcal{F}} \left| \frac{2}{n} \sum_{i=1}^n \sigma_i f(x_i) \right| \right) \quad (2.23)$$

where $\sigma_1, \dots, \sigma_n$ are independent uniform $\{\pm 1\}$ -values random variables. Then the Rademacher complexity is:

$$\mathcal{R}(\mathcal{F}) = \mathbf{E}_X \hat{\mathcal{R}}_n(\mathcal{F}) \quad (2.24)$$

It can be shown the following important result:

Theorem 2.1.3. Let \mathcal{F} be a set of $\{\pm 1\}$ -valued functions, a $0 - 1$ loss function defined on X , and let $(\mathbf{x}_i, y_i)_{i=1}^n$ be training samples:

1. With probability at least $1 - \delta$ every function $f \in \mathcal{F}$ satisfies:

$$R[f] \leq R_{emp}[f] + \mathcal{D}_n(\mathcal{F}) + 3\sqrt{\frac{\ln(1/\delta)}{2n}} \quad (2.25)$$

2. With probability $1 - \delta$ every function $f \in \mathcal{F}$ satisfies:

$$R[f] \leq R_{emp}[f] + \frac{\mathcal{R}_n(\mathcal{F})}{2} + \sqrt{\frac{\ln(1/\delta)}{2n}} \quad (2.26)$$

The last bound can be also expressed in terms of $\hat{\mathcal{R}}_n(\mathcal{F})$:

$$R[f] \leq R_{emp}[f] + \frac{\hat{\mathcal{R}}_n(\mathcal{F})}{2} + 3\sqrt{\frac{\ln(1/\delta)}{2n}} \quad (2.27)$$

Interestingly both (2.25) and (2.27) depends on the sample X ; that is the complexity terms $\hat{\mathcal{D}}$ and $\hat{\mathcal{R}}$ can be estimated using the sample X , thus are sample dependent complexities; this contrasts with d_{vc} that was given by a worst

case analysis and that was sample independent. This fact, intuitively, suggests that MD and Rademacher bounds can be tighter than that based on d_{vc} ; indeed this is the case, and it can be shown that:

$$\mathcal{R}_n(\mathcal{F}) \leq \sqrt{2 \frac{\log |\mathcal{F}|}{n}} \quad (2.28)$$

Thus Rademacher (and MD) bounds are tighter, and so more reliable in giving an estimation of the generalization error.

Consistently with the fact that these bounds are data dependent, they allow to be computed by data-dependent procedures. For the MD bound it can be shown that [26] that a way of computing $\hat{\mathcal{D}}$ is randomly swapping the sign of half of the training set and then using the learning machine in usage to minimize the empirical risk on this new set. Called $R_{emp}[f, \hat{y}]$ the empirical error on the new set of labels \hat{y} then the following holds true:

$$\hat{\mathcal{D}}(\mathcal{F}) = 1 - 2R_{emp}[f, \hat{y}] \quad (2.29)$$

In the usual practice, in order to make robust estimations of $\hat{\mathcal{D}}(\mathcal{F})$, one usually performs several random swaps of \hat{y} that is one computes:

$$\hat{\mathcal{D}}(\mathcal{F}) = 1 - 2 \sum_{i=1}^{it} R_{emp}[f, \hat{y}_i] \quad (2.30)$$

where it is the number of Montecarlo iterations.

For Rademacher complexity the procedure is analogous but now the instead of a random swap of y , directly that random labels σ are used; thus:

$$\frac{\hat{\mathcal{R}}(\mathcal{F})}{2} = 1 - 2 \sum_{i=1}^{it} R_{emp}[f, \sigma_i] \quad (2.31)$$

These bounds are data-dependent bounds, meaning that the complexity term of the class of functions \mathcal{F} can be computed by using training data. Conversely the class of functions \mathcal{F} must be still defined a priori and is not data dependent; it can be shown [30] that a further correction to the generalization bounds must be performed with a fourth term that accounts for the choice of a data dependent class of functions.

When using Biased regularization one is constraining the class of functions

by a data-dependent term; so, in theory, one should use a bound such that presented in [30]; in this work, although not fully theoretically justified, MD bound will be used because the aim is effective model selection and showing that a shrinking of the complexity term can be accomplished; this outcome is up to constant terms that are part of the bounds; moreover the developed framework in [30] seems not to provide an easy way to compute this missing term.

2.1.1.2 PAC-Bayesian Bounds

PAC Bayesian bounds [31] are attractive tools to bound the generalization error of a learning system; in this theory there is a prior on the space of functions; this is opposite to Vapnik theory where the space of functions is flat. This is a strong drawback of Vapnik theory where, instead, there is no ordering and so there is no a priori preference on the space of functions.

In this kind of bounds, given a distribution $D(X)$ of patterns \mathbf{x} lying in a certain input space X , given a distribution Q of classifiers f one defines the true error as $Q_D = \mathbf{E}_f R[f]$ as the probability of misclassifying an instance \mathbf{x} from $D(X)$ with a classifier f chosen according to Q ; and the empirical error $\hat{Q}_S = \mathbf{E}_f f_n$ as the probability of classifier f chosen according to Q misclassifying an instance \mathbf{x} from a sample S . Then it holds the following result:

Theorem 2.1.4. *For all priors distributions $P(f)$ over the classifiers f , and for any $\delta \in (0, 1]$*

$$Pr_{S \sim D^n} \left(\forall Q(f) : KL(\hat{Q}_S \| Q_D) \leq \frac{KL(Q(f) \| P(f)) + \ln \frac{m+1}{\delta}}{n} \right) \geq 1 - \delta \quad (2.32)$$

where KL is the Kullback-Leibler divergence and $KL(Q(f) \| P(f)) = \mathbf{E}_f \ln \frac{Q(f)}{P(f)}$

This bound can be further specified for linear classifiers. For any vector \mathbf{w} one can define a stochastic classifier in the following way: one chooses the distribution $Q(\mathbf{w}, \nu)$ to be a spherical Gaussian with identity covariance matrix centered on the direction given by \mathbf{w} at a distance ν from the origin. One chooses the prior $P(f)$ to be a spherical Gaussian with identity covariance matrix centered on the origin. For classifiers of the form:

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^t \phi(\mathbf{x})) \quad (2.33)$$

the following result holds:

Theorem 2.1.5. *For all distributions D , for all classifiers given by \mathbf{w} and $\nu > 0$, for all $\delta \in (0, 1]$:*

$$Pr_{S \sim D^n} \left(KL(\hat{Q}_S(\mathbf{w}, \nu) \| Q_D(\mathbf{w}, \nu)) \leq \frac{\frac{\nu^2}{2} + \ln \frac{m+1}{\delta}}{n} \right) \geq 1 - \delta \quad (2.34)$$

Moreover it can be shown that:

$$\hat{Q}_S(\mathbf{w}, \nu) = \mathbf{E}^n[\hat{F}(\nu\gamma(\mathbf{x}_i, y_i))] \quad (2.35)$$

where \mathbf{E} is the average over the n patterns and:

$$\gamma(\mathbf{x}_i, y_i) = \frac{y_i \mathbf{w}^t \phi(\mathbf{x})}{\|\phi(\mathbf{x})\| \|\mathbf{w}\|} \quad (2.36)$$

is the normalized margin. $\hat{F} = 1 - F$ where F is the cumulative normal distribution:

$$F(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx \quad (2.37)$$

This bound is quite tight [31] and very elegantly links the margin to the complexity term.

2.2 Learning as a function approximation problem

The problem of learning from examples can be seen also as a function approximation problem [32].

Given n couples of (sample,label) as (\mathbf{x}_i, y_i) where $\mathbf{x}_i \in R^d$ and $\in X$ and y_i is either a binary label $\{-1, +1\}$ or a real value, the problem of learning a function $f(\mathbf{x}, \mathbf{w})$ that predicts y consists in estimating the coefficients $w_j \in R$ of a series expansion using as base functions $\phi(\mathbf{x})$:

$$f(\mathbf{x}, \mathbf{w}) = \sum_{j=1}^m w_j \phi(\mathbf{x}) \quad (2.38)$$

When $\phi(\mathbf{x}) = x^j$ (where x^j specifies the j -th component of the vector \mathbf{x}) and $n = d$ than one gets a linear model of the data:

$$f(\mathbf{x}, \mathbf{w}) = \sum_{j=1}^d w_j x^j \quad (2.39)$$

This model is typical for instance of the perceptron algorithm [33]. When $\phi(\mathbf{x})$ is non linear one gets a non-linear model; this model of the data can be generalized to *multilayers* structures. In this case one expands also the inner functions :

$$f(\mathbf{x}, \mathbf{w}, \hat{\mathbf{W}}, \dots) = \sum_{j=1}^m w_j \phi \left(\sum_{k=1}^o \hat{\mathbf{W}}_{jk} \phi(\dots) \right) \quad (2.40)$$

This expansion type is typical of multilayer neural networks; when the expansion is confined to two levels then the coefficients w_j are called coefficients of the output layer and \hat{w}_{jk} are coefficients of the *hidden* layer.

$$f(\mathbf{x}, \mathbf{w}, \hat{\mathbf{W}}) = \sum_{j=1}^m w_j \phi \left(\sum_{k=1}^d \hat{\mathbf{W}}_{jk} x^k \right) \quad (2.41)$$

A fundamental result given by Cybenko [34] and further generalized by Hornik [35] states that such two-layers networks in a wide class of *activation functions* $\phi(\mathbf{x})$ and with finite m can approximate any function f . This result is important because states that such a structure has a notable *representation* capability. Nothing is said about the generalization capability of these networks and nothing is said about the learning strategy that is needed to learn

the coefficients vector \mathbf{w} and the matrix $\hat{\mathbf{W}}$. The crucial point here is that one needs an algorithm that efficiently learns both \mathbf{w} and $\hat{\mathbf{W}}$ and that grants generalization.

The solution of the first issue historically comes from back-propagation algorithm [36] where sigmoidal activation functions $\phi(\mathbf{x})$ were used; this algorithm is able to minimize functional of the following form:

$$\mathfrak{N}(\mathbf{X}, \mathbf{y}; \mathbf{w}, \hat{\mathbf{W}}) = \sum_{i=1}^n \left(\sum_{j=1}^m w_j \phi \left(\sum_{k=1}^d \hat{\mathbf{W}}_{jk} \mathbf{x}_i^k \right) - y_i \right)^2 \quad (2.42)$$

or:

$$\mathfrak{N}(\mathbf{X}, \mathbf{y}; \mathbf{w}, \hat{\mathbf{W}}) = \sum_{i=1}^n \left(\phi \left(\sum_{j=1}^m w_j \phi \left(\sum_{k=1}^d \hat{\mathbf{W}}_{jk} \mathbf{x}_i^k \right) \right) - y_i \right)^2 \quad (2.43)$$

The BP algorithm, now, is only one of the possible choices to minimize such functionals; other methods can be used such the Levenberg-Marquardt method [37]. In order to cope with generalization problems two key quantities play an important role: the number of hidden units m and the number of available samples at learning time n . The first parameter determines how much one choose to fit the seen data; intuitively for an high number of hidden units then by representations theorem one is able to certainly fit, and interpolate, the given data, but still nothing is assured on the prediction capability of the network, that is one can overfit the data; conversely with a low number of hidden units one can be unable to fit the data, thus oversmoothing. Moreover one does not know how much the seen samples \mathbf{x}_i are representative of the entire population and how much noise is over them. Different studies have inquired on what is a suitable strategy that can deal with such issues; probably one of the most important work [38] shown that, what it counts to get a proper generalization capability, is not the number of hidden units but the norm, that is, the size of the weights themselves. This aspect gives a link to the classical theory of Regularization [39] where to stabilize an inverse problem solution the norm of the weights is bounded; this lead to the conclusion that learning is, at least formally, an inverse problem [40]. Mathematically speaking, instead of minimizing a square loss, one minimizes a regularized

square loss, where weights size is minimized simultaneously:

$$\mathfrak{N}(\mathbf{X}, \mathbf{y}; \mathbf{w}, \hat{\mathbf{W}}, \lambda_1, \lambda_2) = \sum_{i=1}^n \left(\phi \left(\sum_{j=1}^m w_j \phi \left(\sum_{k=1}^d \hat{\mathbf{W}}_{jk} \mathbf{x}_i^k \right) \right) - y_i \right)^2 + \lambda_1 \|\hat{\mathbf{W}}\|_F^2 + \lambda_2 \|\mathbf{w}\|^2 \quad (2.44)$$

where $\|\cdot\|_F$ indicates the Frobenius norm and λ_1, λ_2 are regularization parameters that control the tradeoff between regularization and fitting. So far the problem has been only moved from the proper size of weights, to the proper values of λ_1, λ_2 . Regularization theory mainly tends to use regularization as a mathematical stabilizer, conversely in learning theory, the computational stabilization is not the only issue and neither it is the most important, here the aim is prediction, that is generalization; thus a proper choice of the regularization parameter is critical.

The network topology so far exposed has the algorithmic drawback that the cost function $\mathfrak{N}(\mathbf{X}, \mathbf{y}; \mathbf{w}, \hat{\mathbf{W}}, \lambda_1, \lambda_2)$ is not convex when a two-layer structure is used due to the non convexity of the activations functions $\phi(x)$; whenever one gets a solution, this solution is only local and not unique. Another problem that concerns these networks is that one can experimentally find that are not particularly amenable to deal with, the so called *curse of dimensionality*. Summarizing these kind of networks are powerful approximating tools however suffer from the above exposed problems; for these reasons in the last years alternative models have been proposed that still endow regularized costs; these methods are kernel methods.

2.2.1 Reproducing Kernel Hilbert Spaces

Kernel Methods are a relatively recently introduced class of methods that have the main advantages to mitigate the problem of the curse of dimensionality and that consists in convex or strictly convex functionals.

Historically the Support Vector Machine algorithm (SVM) [3] has been the first introduced Kernel Machine; then on the basis of SVM the SVM functional has been generalized [41] to a wide class of learning algorithms. In order to explain the rationale behind this category of algorithms one needs some maths on the notions of Reproducing Kernel Hilbert Spaces (RKHS)

[41], that are the spaces where the learned function f lives.

A Reproducing Kernel Hilbert Space (RKHS) is a Hilbert space \mathcal{H} of functions defined over some bounded domain $\Omega \subset R^d$ with the property that, for each $\mathbf{x} \in \Omega$, the evaluation functionals $\mathcal{F}_{\mathbf{x}}$ defined as:

$$\mathcal{F}_{\mathbf{x}}[f] = f(\mathbf{x}) \forall f \in \mathcal{H} \quad (2.45)$$

are linear, bounded functionals. The boundedness means that there exists a $U = U_{\mathbf{x}} \in R^+$ such that:

$$|\mathcal{F}_{\mathbf{x}}[f]| = |f(\mathbf{x})| \leq U \|f\| \quad (2.46)$$

for all f in the RKHS. It can be proved [41] that to every RKHS \mathcal{H} there corresponds a unique positive definite function $K(\mathbf{x}, \mathbf{y})$ of two variables in X , called the reproducing kernel of \mathcal{H} , that has the following reproducing property:

$$f(\mathbf{x}) = \langle f(\mathbf{y}), K(\mathbf{y}, \mathbf{x}) \rangle_{\mathcal{H}} \forall f \in \mathcal{H} \quad (2.47)$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the scalar product in \mathcal{H} . The function K behaves in \mathcal{H} as the delta function does in L_2 , although L_2 is not a RKHS because not all the functionals $\mathcal{F}_{\mathbf{x}}$ are bounded. It exists a relatively simple way to construct kernels [41]: let assume that one has a sequence of positive numbers γ_j and linearly independent functions $\phi_j(\mathbf{x})$ such that they define a function $K(\mathbf{x}, \mathbf{y})$ in the following way :

$$K(\mathbf{x}, \mathbf{y}) = \sum_{j=0}^{\infty} \gamma_j \phi_j(\mathbf{x}) \phi_j(\mathbf{y}) \quad (2.48)$$

where the series is well defined (for example it converges uniformly). A simple calculation [41] shows that the function K defined in the previous equation is positive definite. Now assume that the functions that belong to \mathcal{H} has the following model:

$$f(\mathbf{x}) = \sum_{j=0}^{\infty} w_j \phi_j(\mathbf{x}) \quad (2.49)$$

for any $w_j \in R$ and define the scalar product to be:

$$\langle \sum_{j=0}^{\infty} w_j \phi_j(\mathbf{x}), \sum_{j=0}^{\infty} v_j \phi_j(\mathbf{x}) \rangle_{\mathcal{H}} = \sum_{j=0}^{\infty} \frac{w_j v_j}{\gamma_j} \quad (2.50)$$

Such a space is a RKHS; it is sufficient to check that the reproducing property holds:

$$\langle f(\mathbf{y}), K(\mathbf{y}, \mathbf{x}) \rangle_{\mathcal{H}} = \sum_{j=0}^{\infty} \frac{w_j \gamma_j \phi_j(\mathbf{x})}{\gamma_j} = \sum_{j=0}^{\infty} w_j \phi_j(\mathbf{x}) = f(\mathbf{x}) \quad (2.51)$$

When one has a finite number of basis ϕ_j , the γ_j can be arbitrary finite numbers, since convergence is ensured; in particular they can be all equal to one. Generally, it is easy to show [41] that whenever a function K of the form (2.48) is available, it is possible to construct a RKHS as shown above. Vice versa, for any RKHS there is a unique kernel K and corresponding γ_j, ϕ_j , that satisfy (2.49) and for which equations the defined dot product hold for all functions in the RKHS. Moreover, equation (2.51) shows that the norm of the RKHS has the form:

$$\|f\|_{\mathcal{H}}^2 = \sum_{j=0}^{\infty} \frac{w_j^2}{\gamma_j} \quad (2.52)$$

The ϕ_j are a basis for the RKHS (not necessarily orthonormal), and the kernel K is the *correlation* matrix associated with these basis functions. The space $\{(\phi_j(\mathbf{x}))_{j=0}^{\infty}, \mathbf{x} \in X\}$ is called the *feature space* induced by the kernel K .

2.2.1.1 RKHS and Regularization Operators

A fundamental property links operator theory and kernels; the Mercer's theorem [42] states that any function $K(\mathbf{x}, \mathbf{y})$ which is the kernel of a positive operator in $L_2(\Omega)$ has an expansion of the form (2.48), in which the ϕ_j and the γ_j are respectively the orthogonal eigenfunctions and the positive eigenvalues of the operator corresponding to K . In [43] it is reported that the positivity of the operator associated to K is equivalent to the statement that the kernel K is positive definite, that is the matrix $\mathbf{K}, K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ is positive definite for all choices of distinct points $\mathbf{x}_i \in X$. However to build a kernel it is sufficient that ϕ_j are linearly independent and are not necessarily eigenfunctions. Summarizing the features of kernel functions and RKHS the following properties hold:

1. There exists a unique element $K(\mathbf{x}, \mathbf{y})$ for each $x \in X$ such that: $f(\mathbf{x}) = \langle K(\mathbf{x}, \mathbf{y}), f(\mathbf{y}) \rangle_{\mathcal{H}}$, for all $f \in \mathcal{H}$; this is the reproducing property

2. The function $K(\mathbf{x}, \mathbf{y})$ is positive definite kernel functions
3. Any inner product $\langle g, f \rangle_{\mathcal{H}}$ can be uniquely expressed in the form $f^t T g$ where T is a positive definite operator
4. $K_{ij} = T_{ij}^{-1}$ or equivalently $K = T^{-1}$
5. The kernel \mathbf{K} defines the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ uniquely.

This set of features imply that for a given covariance matrix K one can define a RKHS by setting:

$$\langle f, g \rangle_{\mathcal{H}} = f^t \mathbf{K}^{-1} g \quad (2.53)$$

Given this property one can set a link with regularization operators; if one defines a one-to-one linear regularization operator \mathbf{R} and use as kernel $\mathbf{K} = (\mathbf{R}^t \mathbf{R})^{-1}$ then one gets:

$$\|f\|_{\mathcal{H}}^2 = f^t \mathbf{K}^{-1} f = f^t \mathbf{R}^t \mathbf{R} f = \|\mathbf{R}f\|^2 \quad (2.54)$$

That is, if $\|\mathbf{R}f\|$ measures the *regularity* of f then the RKHS norm exactly matches the regularity measure. This setting corresponds to asking for functions that fulfill the model equation, or regularity condition:

$$\mathbf{R}f = 0 \quad (2.55)$$

If \mathbf{R} is assumed one-to-one only the null function f can exactly respect the above equation. This properties suggests that kernels can be derived from regularization operators; however commonly used kernels can be defined in closed form, examples are:

1. $K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$, dot product, linear kernel
2. $K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$, Gaussian Radial Basis functions
3. $K(\mathbf{x}, \mathbf{y}) = (\|\mathbf{x} - \mathbf{y}\|^2 + c^2)^{1/2}$, multiquadric
4. $K(\mathbf{x}, \mathbf{y}) = (\|\mathbf{x} - \mathbf{y}\|^2 + c^2)^{-1/2}$, Inverse multiquadric
5. $K(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^{2n+1}$, or $K(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^{2n} \ln(\|\mathbf{x} - \mathbf{y}\|)$, thin plate splines

6. $K(\mathbf{x}, \mathbf{y}) = (1 + \langle \mathbf{x}, \mathbf{y} \rangle)^p$, polynomial of degree p

Every kernel induces a different feature space; this space can be infinite dimensional or not. In the case of an homogeneous polynomial kernel of degree p and $\mathbf{x} \in R^d$ it can be shown [44] that the dimension of the feature space is equal to:

$$\binom{d+p-1}{p} \tag{2.56}$$

thus the feature space dimension grows very quickly with respect to the original space dimension d .

Differently the Gaussian RBF kernel induces an infinite dimensional feature space. This last result is particularly important; this fact, in the mono-dimensional case, can be proved observing the following relations:

$$K(x, y) = e^{-\gamma(x-y)^2} \tag{2.57}$$

$$= e^{-\gamma x^2 + 2\gamma xy - \gamma y^2} \tag{2.58}$$

$$= e^{-\gamma(x^2+y^2)} \left(1 + \sqrt{\frac{2\gamma}{1!}} x \sqrt{\frac{2\gamma}{1!}} y + \sqrt{\frac{(2\gamma)^2}{2!}} x^2 \sqrt{\frac{(2\gamma)^2}{2!}} y^2 + \sqrt{\frac{(2\gamma)^3}{3!}} x^3 \sqrt{\frac{(2\gamma)^3}{3!}} y^3 + \dots \right) \tag{2.59}$$

$$= \phi(x)^t \phi(y) \tag{2.60}$$

thus:

$$\phi(x)^t = e^{-\gamma x^2} \left[1, \sqrt{\frac{2\gamma}{1!}} x, \sqrt{\frac{(2\gamma)^2}{2!}} x^2, \sqrt{\frac{(2\gamma)^3}{3!}} x^3, \dots \right] \tag{2.61}$$

where due to the Taylor expansion the induced space is infinite dimensional. Another suggestive relation that holds for the gaussian kernel regards its link with its underlying regularization operator: the proof outline that now is given follows that in [45].

Assume \mathcal{X} to be the discretized real line $\mathcal{X} = \{i/h, i = 1, \dots, N\}$ and let $L(\theta) = \sum_{i=0}^n a_i \theta^i$ be a n-th order polynomial. One can consider the linear ODE:

$$L(\mathcal{D})[f] = \sum_{i=0}^n a_i \mathcal{D}^i f = 0 \tag{2.62}$$

where \mathcal{D} is the first derivative operator and $f : \mathcal{X} \rightarrow R$. Then assume periodic boundary conditions so one can use the Fourier transform to express

the derivative operator, moreover assume that $L(\mathcal{D})$ is a one-to-one operator. By such conditions the operator \mathcal{D} can be approximated by the matrix (i.e. $N = 5$ case):

$$\frac{1}{h} \begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & 0 & -1 \end{pmatrix} \quad (2.63)$$

Then \mathcal{D} can be diagonalized in the Fourier basis, $\mathcal{D} = \sum_{k=1}^N u_k w_k u_k^t$ where $w_k = \frac{1}{h} \exp(i(2\pi/N)k) - 1$ where i is the imaginary unit and $\delta_{x_j}^t u_k = \exp(i(2\pi/N)jk)$ where $\delta_{x_j} = \delta_{ij}$ for $j = 1, \dots, N$. Thus the operator $L(\mathcal{D})$, that is the regularization operator, can be discretized in the Fourier basis; then using the fact that $K = (R^t R)^{-1}$ and the fact that operations on $L(\mathcal{D})$ are equivalent to operations of its eigenvalues, one can obtain the associated kernel:

$$K(x_l, x_m) = (L(\mathcal{D})^t L(\mathcal{D}))_{lm}^{-1} \quad (2.64)$$

$$= \delta_{x_l}^t (\overline{L(\mathcal{D})}^t L(\mathcal{D}))^{-1} \delta_{x_m} \quad (2.65)$$

$$= \sum_{k=1}^N \delta_{x_l}^t u_k \frac{1}{\overline{L(w_k)}^t L(w_k)} u_k^t \delta_{x_m} \quad (2.66)$$

$$= \sum_{k=1}^N \frac{1}{|L(w_k)|^2} \exp\left(i \frac{2\pi}{N} k(l - m)\right) \quad (2.67)$$

Thus, the kernel $K(x, y)$ is the discrete Fourier transform of $g(w_k) = 1/|L(w_k)|^2$; since g is real-valued, the Fourier transform of it is also real and symmetric; the corresponding kernel is real valued and only depends on the distance between x_l and x_m , thus $K(x_l, x_m) = K(x_l - x_m)$ is translation invariant.

Such a result can be motivated from the regularization point of view: high derivatives are described by polynomials $L(\theta)$ of high order, in which case $\|L(\mathcal{D})f\|^2 = \sum_k f^t u_k |L(w_k)|^2 u_k^t f$ strongly penalizes high frequencies; thus the corresponding kernel then contains few high frequency components, hence is relatively smooth.

The reverse derivation, from a translation invariant kernel function on \mathcal{X} to a differential regularization operator is interesting too: in order to derive a DE, invert the eigenvalues of the kernel matrix K , take the square root and interpolate the result by a polynomial L of degree at most N . The obtained poly-

nomial coefficients a_i gives the coefficients of the linear operator in (2.62). An interesting property holds for the Gaussian kernel $K(x_i, x_j) = e^{-|i-j|^2/(2\sigma^2)}$; its discrete Fourier transform is difficult to compute analytically so one approximates it with its continuous version for large N and small $1/h$ step size. It is well known that the continuous Fourier transform of a Gaussian is again a Gaussian with variance σ^{-2} . Inverting and taking the square root one derives the function $\exp((\sigma^2/4)w^2)$ whose Taylor expansion is $L(w) = \sum_{n=0}^{\infty} (\sigma^{2n}/2^{2n}n!)w^{2n}$. Replacing w with the derivative ∂_x one gets:

$$L(\mathcal{D}) = \sum_{n=0}^{\infty} \frac{\sigma^{2n}}{2^{2n}n!} \mathcal{D}^{2n} \quad (2.68)$$

That is the Gaussian kernel is equivalent to regularization with derivatives of all even orders. A large σ leads to strong penalization of high derivatives, i.e. to smoother functions; conversely small σ leads to less smooth functions. This result can be seen and generalized to all kernels using the discrete point of view. Consider the kernel matrix \mathbf{K} induced by any kernel. Suppose to compute the SVD and to express \mathbf{K} as:

$$\mathbf{K} = \sum_{j=1}^n \hat{\gamma}_j \mathbf{u}_j \mathbf{u}_j^t \quad (2.69)$$

where $\hat{\gamma}_j$ are the singular values and \mathbf{u}_j are the eigenvectors. This equation can be seen as the discrete version of (2.48) where \mathbf{u}_j are the discretized version of $\phi(\mathbf{x})_j$. Then by (2.52) one gets:

$$\|f\|_{\mathcal{H}}^2 = \sum_{j=1}^n \frac{w_j^2}{\hat{\gamma}_j} \quad (2.70)$$

That is singular values of low values (i.e. noise, i.e. high frequencies) generates an high penalization, conversely high values does not penalize. Thus for this reason when one refers to kernel methods, one often refers to *spectral methods* [46], because regularization, and thus generalization, comes from the modification of the spectrum of the kernel matrix.

2.2.1.2 RKHS and Neural Networks

Given these premises on the properties of RKHS one now can relate the model of neural networks and the model given by RKHS; in the first case for a two-layer network the model is:

$$f(\mathbf{x}) = \sum_{j=1}^m w_j \phi \left(\sum_{k=1}^d \hat{\mathbf{W}}_{jk} \mathbf{x}^k \right) \quad (2.71)$$

instead in a RKHS the model f (2.49) can be written as:

$$f(\mathbf{x}) = \sum_{j=1}^m w_j \phi_j(\mathbf{x}) \quad (2.72)$$

Interestingly in the first case one needs to learn the weights w and the matrix $\hat{\mathbf{W}}$; in the second case one needs to only learn the vector of weights w ; as it will be later shown in RKHS learning is possible also in the case $\phi(\mathbf{x})$ is infinite dimensional; this is the case of gaussian kernel. Thus one is able to understand how much are powerful RKHS: one can implicitly use a infinite dimensional space for representing $f(\mathbf{x})$ but can still evaluate it by a finite computations; this property is often called the *kernel trick*.

The counterpart of this fact is that machines based on fixed kernel functions does not learn the kernel itself; neural networks try to learn the matrix $\hat{\mathbf{W}}$ hence try to learn the similarities between patterns; kernel based method *does not learn the similarities* between couples of patterns, thus a kernel machine, essentially, is as powerful as its kernel representation is.

2.2.1.3 Representer theorem

A central result about RKHS is the above mentioned *Representer Theorem* [47]; the result presented here is a slight generalization [48]:

Theorem 2.2.1. (*Representer Theorem*) Suppose having a non empty set X , a positive definite real-valued kernel function K on $X \times X$, a training set $(\mathbf{x}_i, y_i) \in X \times R$ $i = 1, \dots, n$, a strictly monotonically increasing real-valued function g on $[0, \infty)$, an arbitrary loss function $L : (X \times R^2)^n \rightarrow R \cup \{\infty\}$, and

a class of functions \mathcal{F} of a RKHS, then any $f \in \mathcal{F}$ minimizing the regularized risk functional:

$$\sum_{i=1}^n L(f_i, y_i) + \lambda g(\|f\|_{\mathcal{H}}^2) \quad (2.73)$$

admits a representation of the form :

$$f(\mathbf{x}) = \sum_{i=1}^n \beta_i K(\mathbf{x}, \mathbf{x}_i) \quad (2.74)$$

When dealing with most kernel machines the function $g(x) = x$ is used thus leading to the functional:

$$\mathfrak{N}(X, y; f, \lambda) = \sum_{i=1}^n L(f_i, y_i) + \lambda \|f\|_{\mathcal{H}}^2 \quad (2.75)$$

Depending on the loss $L(f_i, y_i)$ different machines can be obtained; figure (2.1) depicts the following loss functions for classification:

1. If $L(f_i, y_i) = (1 - y_i f_i)_+$ where $(\cdot)_+ = \max(0, \cdot)$ then one gets a Support Vector Machine for classification.
2. If $L(f_i, y_i) = (y_i - f_i)^2$ one gets Regularized Least Squares, also known as Kernel Ridge Regression, or the kernelized version of Tikhonov regularization.
3. If $L(f_i, y_i) = \log(1 + e^{-y_i f_i})$ one gets Kernel Logistic Regression.

Representer theorem holds for every loss functions thus leading to a family of learning algorithms; despite this degree of freedom, usually convex loss functions are used, because in this way the entire cost function $\mathfrak{N}(\mathbf{X}, y; f, \lambda)$ is convex and a global minimizer f is obtained; all the presented loss functions are convex.

Recently different generalization of kernel machines were proposed, these are: kernel methods for multi-task learning [49], kernel methods for embedding probability distributions [50], multiple kernel learning algorithms [51] and kernel machines based on Reproducing Kernel Krein Spaces [52]. The

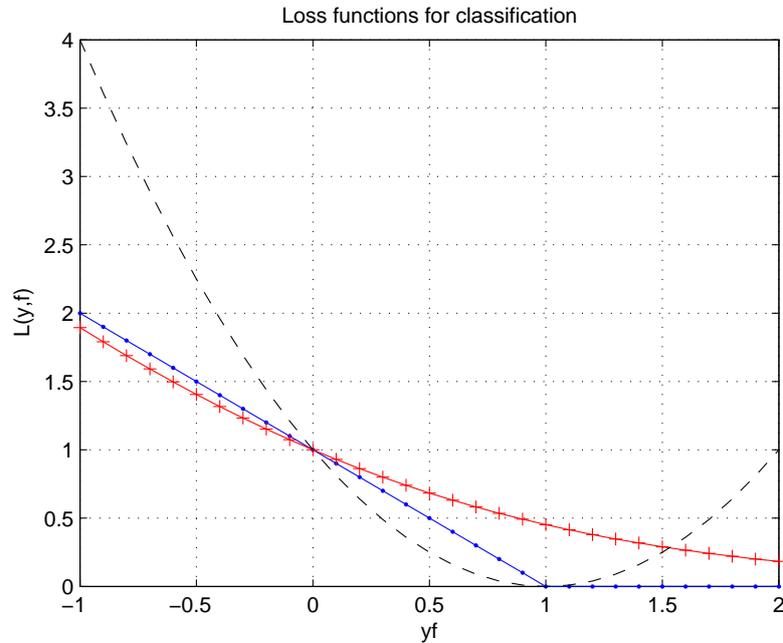


Figure 2.1: Loss functions for classification: blu dotted line is hinge loss, red '+' line is logistic loss, black dashed line is square loss

last two are used in this thesis. In multiple kernel learning one wants to recover the ability of neural networks to learn similarities; the setting consists in defining a kernel as convex superposition of different kernels and then learning the weights of this convex combination; this tool will be used later for feature selection for the mean estimation section. In Reproducing Kernel Krein Spaces, instead of using positive definite kernels one uses indefinite kernels; such kernel can often arise in specific learning domains where defining a positive definite kernel is not natural.

2.2.1.4 Biased Regularization

An important extension, that will be discussed throughout this thesis, of the functional (2.75) is the case in which regularization is performed around a reference model f_0 . Such functional is:

$$\sum_{i=1}^n L(f_i, y_i) + \lambda \|f - f_0\|_{\mathcal{H}}^2 \quad (2.76)$$

or an even more generalized form as:

$$\sum_{i=1}^n L(f_i, y_i) + \lambda_1 \|f - \lambda_2 f_0\|_{\mathcal{H}}^2 \quad (2.77)$$

the rationale behind this generalization will be clear when discussing biased regularization. One can anticipate that such models still admit a representation theorem in which the learned function both depends on training data and on f_0 .

The following sub-sections describe three common kernel methods that will be used and enriched in the following sections: these machines are Regularized Least Squares, Support Vector Machines and Kernel Logistic Regression. Then the unsupervised learning paradigm and other learning tools will be introduced.

2.2.2 Kernel Methods

This section presents some classical and widely used learning methods, namely, Regularized Least Squares [53], Support Vector Machines [3] and Kernel Logistic Regression [54]. These methods have in common the functional (2.75) but use different loss functions, and consequently exhibit different computational and model features. In this section the matrix \mathbf{X} will denote the matrix of n points $x_i \in \mathbb{R}^d$, \mathbf{w} will denote the set of weights of a linear model $\mathbf{f} = \mathbf{X}\mathbf{w}$, y_i is the label of each x_i and \mathbf{y} denotes the vector of y_i

2.2.2.1 Regularized Least Squares

Regularized Least Squares (RLS) is a very simple learning algorithm both from the conceptual and algorithmic point of view; one can start from the linear version and then study the kernel counterpart.

The linear version of RLS is the well known Tikhonov regularization method [39]. In Tikhonov regularization one employs a square loss and a linear model $\mathbf{f} = \mathbf{X}\mathbf{w}$. The functional to be minimized with respect to w is:

$$\mathfrak{N}(\mathbf{X}, \mathbf{w}, \mathbf{y}; \lambda) = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2 \quad (2.78)$$

where λ represents the regularization parameter. The solution of this problem is easily obtained by setting $\nabla_{\mathbf{w}} \aleph(\mathbf{X}, \mathbf{w}, \mathbf{y}; \lambda) = 0$; the result is that one has to solve the following system of equations:

$$(\mathbf{X}^t \mathbf{X} + \lambda \mathbf{I}_{dd}) \mathbf{w} = \mathbf{X}^t \mathbf{y} \quad (2.79)$$

where \mathbf{I}_{dd} is a $d \times d$ identity matrix.

To deal with its non-linear counterpart one has to generalize the functional to a RKHS. Then one has:

$$\aleph(\mathbf{f}, \mathbf{y}; \lambda) = \|\mathbf{f} - \mathbf{y}\|^2 + \lambda \|\mathbf{f}\|_{\mathcal{H}}^2 \quad (2.80)$$

By using the relation between kernel and regularization operators, one can rewrite the functional as:

$$\aleph(\mathbf{f}, \mathbf{y}; \lambda) = \|\mathbf{f} - \mathbf{y}\|^2 + \lambda \mathbf{f}^t \mathbf{R}^t \mathbf{R} \mathbf{f} \quad (2.81)$$

By using the representer theorem one knows that $f(\mathbf{x}) = \sum_{i=1}^n \beta_i K(\mathbf{x}, \mathbf{x}_i)$; moreover recalling that $\mathbf{K} = (\mathbf{R}^t \mathbf{R})^{-1}$ one gets:

$$\aleph(\boldsymbol{\beta}, \mathbf{y}; \lambda) = \|\mathbf{K} \boldsymbol{\beta} - \mathbf{y}\|^2 + \lambda \boldsymbol{\beta}^t \mathbf{K} \mathbf{K}^{-1} \mathbf{K} \boldsymbol{\beta} \quad (2.82)$$

that finally is:

$$\aleph(\boldsymbol{\beta}, \mathbf{y}; \lambda) = \|\mathbf{K} \boldsymbol{\beta} - \mathbf{y}\|^2 + \lambda \boldsymbol{\beta}^t \mathbf{K} \boldsymbol{\beta} \quad (2.83)$$

Again setting the gradient to zero, $\nabla_{\boldsymbol{\beta}} \aleph(\boldsymbol{\beta}, \mathbf{y}; \lambda) = 0$ lead to the following linear system:

$$(\mathbf{K} + \lambda \mathbf{I}_{nn}) \boldsymbol{\beta} = \mathbf{y} \quad (2.84)$$

The formulation (2.83) generalizes the linear case: setting the kernel as linear kernel $\mathbf{K} = \mathbf{X} \mathbf{X}^t$ one gets:

$$\aleph(\mathbf{f}, \mathbf{y}; \lambda) = \|\mathbf{X} \mathbf{X}^t \boldsymbol{\beta} - \mathbf{y}\|^2 + \lambda \boldsymbol{\beta}^t \mathbf{X} \mathbf{X}^t \boldsymbol{\beta} \quad (2.85)$$

setting $\mathbf{X}^t \boldsymbol{\beta} = \mathbf{w}$ one recovers Tikhonov regularization:

$$\aleph(\mathbf{w}, \mathbf{y}; \lambda) = \|\mathbf{X} \mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2 \quad (2.86)$$

In this problem is evident the action of the regularization; $\lambda \mathbf{I}_{nn}$ acts as shift of the eigenvalues of \mathbf{K} as it was suggested in the previous sections. Importantly

this shift action improves on the algorithmic stability of the learning process because the presence of $\lambda \mathbf{I}_{nn}$ improves on possibly bad conditioning of \mathbf{K} . From this point of view is evident that learning can be seen as an attempt to stabilize and solve an ill-conditioned inverse problem.

In general one can say that exist a whole class of learning algorithm whose learning step is based on an action performed on the spectrum of \mathbf{K} . In other words a spectral algorithm [46] performs as a filter as per:

$$\omega(\mathbf{K})\boldsymbol{\beta} = \mathbf{y} \quad (2.87)$$

Different examples of spectral filtering are possible such as: the ν -method, early stopping and Truncated SVD. The Truncated SVD algorithm first computes the SVD of \mathbf{K} and then re-builds it by setting to 0 all the singular values less than a prescribed threshold λ that acts as regularization parameter; the rebuilt-matrix $\hat{\mathbf{K}}$ then is used to perform learning as per:

$$\hat{\mathbf{K}}\boldsymbol{\beta} = \mathbf{y} \quad (2.88)$$

From the algorithmic point of view RLS solution is obtained by solving the linear system of equations (2.84). This procedure scales as $O(n^3)$ in time, due to Gaussian elimination, and $O(n^2)$ in memory due to kernel matrix. Several approximations methods were proposed to speed-up the learning step of RLS: one is using sparse kernel matrix that allows efficient use of the Conjugate Gradient Method [55], another is using the Nystrom approximation [56]; in this thesis a quite raw approximation method for RLS for classification will be proposed that is particularly simple, fast and amenable for low power devices such as DSPs.

For RLS exists a relatively simple method to compute all the *regularization path* of the learning problem; by regularization path one indicates the set of solutions $\boldsymbol{\beta}_\lambda$ with varying λ . Using the SVD for the kernel matrix one gets $\mathbf{K} = \mathbf{U}\mathbf{S}\mathbf{U}^t$. It is not difficult to show that the solution of RLS can be rewritten as:

$$\boldsymbol{\beta}_\lambda = \mathbf{U}\omega(\mathbf{S})\mathbf{U}^t\mathbf{y} \quad (2.89)$$

where $\omega(\mathbf{S})$ is the filter function for RLS defined by:

$$\omega(\mathbf{S}) = \text{diag} \left(\frac{\gamma_i}{\gamma_i^2 + \lambda} \right) \quad (2.90)$$

where γ_i are the singular values. This fact suggests a fast strategy to compute the set of solutions β_λ : first one computes only once the SVD of \mathbf{K} then by only matrix multiplications one recover all the set β_λ ; in this way for changing λ only one time an SVD is needed, instead of solving each time a linear system. This strategy is effective whenever two conditions are met: first SVD is fast enough such that is computationally convenient over direct solution, secondly the SVD must be stable and reliable. To author experience (experiments are not reported for brevity) the first condition is usually met, conversely the second is much more critical and quite often gaussian elimination solution is considerably more accurate than that obtained by SVD and reconstruction; this holds at least for Matlab environment.

Another point that should be stressed is that, usually, the coefficients vector β is not sparse; this means that when performing a prediction on a new pattern \mathbf{z} , for computing $f(\mathbf{z})$ one needs to compute all the kernel values $K(\mathbf{x}_i, \mathbf{z})$ where i runs over all the n training patterns. This problem, more or less, is common to all kernel machines; to this aim methods such as the *Reduced Set* [57] method have been developed in order to cope with this problem.

2.2.2.2 Support Vector Machines

Support Vector Machine invention has been a breakthrough on the development of learning systems. SVM has a lot of nice features that makes it quite unique:

1. It is very effective in classification problems
2. It is, by far, the most efficient kernel method
3. It has a clear geometrical justification
4. It is theoretically well founded by the Statistical Learning theory point of view

5. Its solution vector β is sparse

Most of its merits derive from its loss function that constitutes a very happy intuition; roughly speaking, in two-class classification problems when $f(\mathbf{x})$ correctly predicts the sign of \mathbf{x} than no errors should be payied; instead when $f(\mathbf{x})$ mismatches with \mathbf{x} true class than a linear error is payied; this description can be formalized by the hinge loss function [3].

The SVM functional in general can be written as:

$$\sum_{i=1}^n (1 - y_i f(\mathbf{x}_i))_+ + \lambda \|f\|_{\mathcal{H}}^2 \quad (2.91)$$

For historical reasons this functional is usually written by using the C parameter $C = 1/\lambda$, substituing the loss function by corresponding linear constraints; moreover the original SVM formulation allows a non regularized bias term b such that the model is $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$ in the linear case and $f(\mathbf{x}) = \langle \mathbf{w}^t, \mathbf{x} \rangle + b$; again SVM is written in its *primal* linear formulation. After these observations the functional is written as the following quadratic programming problem:

$$\begin{aligned} \min_{\mathbf{w}, b, \varepsilon_i} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \varepsilon_i \\ & y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \varepsilon_i \quad \forall i \\ & \varepsilon_i \geq 0 \quad \forall i \end{aligned} \quad (2.92)$$

In the last functional the set of variables ε_i are slack variables that account for the loss function values, and \cdot indicates dot product.

Within this formulation a direct application representer theorem is not amenable; for this reason one can use the convexity of the functional and build its Lagrangian dual form. Denoting by α_i and ξ_i (both > 0 by definition) the introduced Lagrangia multipliers, the Lagrangian \mathcal{L} is:

$$\mathcal{L}(\mathbf{w}, \varepsilon, \alpha, \xi, b) = \frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^n \varepsilon_i - \sum_{i=1}^n \alpha_i [y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \varepsilon_i] - \sum_{i=1}^n \xi_i \varepsilon_i \quad (2.93)$$

The solution of this problem by duality theory [58] is:

$$\max_{\alpha, \xi} \min_{\mathbf{w}, b, \varepsilon_i} \mathcal{L}(\mathbf{w}, \varepsilon, \alpha, \xi, b) \quad (2.94)$$

In order to get the inner minimum one can compute the following partial derivatives:

$$\begin{aligned}\nabla_{\mathbf{w}}\mathcal{L} = 0 &\rightarrow \mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \\ \frac{\partial \mathcal{L}}{\partial b} = 0 &= \sum_{i=1}^n \alpha_i y_i \\ \frac{\partial \mathcal{L}}{\partial \varepsilon_i} = 0 &= C - \alpha_i - \beta_i\end{aligned}\tag{2.95}$$

Considering that $\xi_i, \alpha_i \geq 0$ and that $\xi_i = C - \alpha_i$ this means that the following *box* constraint holds:

$$0 \leq \alpha_i \leq C \quad \forall i\tag{2.96}$$

Substituting the primal variables $\varepsilon, b, \mathbf{w}$ with their corresponding dual, after a few algebraical manipulations, one gets the following quadratic problem:

$$\begin{aligned}\min_{\boldsymbol{\alpha}} \frac{1}{2} \boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha} - \sum_{i=1}^n \alpha_i \\ \sum_{i=1}^n \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C \quad \forall i\end{aligned}\tag{2.97}$$

where \mathbf{Q} is the matrix of elements $q_{ij} = y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)$. The above functional depends on the data X only by dot products; for this reason instead of the dot product one can use any positive definite kernel function, such that the matrix \mathbf{Q} is made up of the elements $q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$.

Using the equality $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$ for linear kernel than one can generalize its non linear counterpart as per:

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \phi(\mathbf{x}_i)\tag{2.98}$$

Being ϕ unknown as usual one cannot have the closed form of \mathbf{w} , that as usual is infinite dimensional; however point-wise evaluations are possible observing that:

$$f(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x}) = \sum_{i=1}^n \alpha_i y_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) + b\tag{2.99}$$

the term $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x})$ is the kernel $K(\mathbf{x}_i, \mathbf{x})$ thus:

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\tag{2.100}$$

That is a representer theorem for SVM where the expansion coefficients are $\alpha_i y_i$.

It can be shown by computing the dual of the dual that the Karush-Kuhn-Tucker (KKT) optimality conditions lead to the complementary slackness conditions:

$$\begin{aligned}\mu_i \alpha_i &= 0 \\ \varepsilon_i (C - \alpha_i) &= 0\end{aligned}\tag{2.101}$$

where $\mu_i \geq 0$ is the slack of the constraint $y_i(\mathbf{w}^t) = 1 + \mu_i - \varepsilon_i$ and $\varepsilon_i \geq 0$ are the Lagrangian multipliers of dual of the dual.

Given these conditions three different cases are possible:

1. $\alpha_i = 0$: then $\mu_i \geq 0$ and $\varepsilon_i C = 0$, thus $\varepsilon_i = 0$; this in turn implies that $y_i f_i \geq 0$, that is the case of no errors. These points do not contribute to the computation of the function $f(\mathbf{x})$.
2. $\alpha_i = C$: then $\mu_i = 0$ and $\varepsilon_i \geq 0$ thus $y_i f_i \leq 0$, that is the case of a wrong prediction. These samples are called Bounded Support Vectors.
3. $0 \leq \alpha_i \leq C$: both $\mu_i = \varepsilon_i = 0$ and $y_i f_i = 1$. These samples are called True Support Vectors.

These three cases show that at the optimum the function $f(\mathbf{x})$ is computed over only the Bounded Support Vectors and the True Support Vectors, thus, given a number n_{sv} of Support Vectors (both true and bounded) the function f is evaluated as:

$$f(\mathbf{x}) = \sum_{i=1}^{n_{sv}} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\tag{2.102}$$

This shows that the solution is sparse and that only some samples *support* the solution. The True Support Vectors are those who lie at a distance from the hyperplane $\mathbf{w}^t \mathbf{x} + b$, called *margin* of value $2/\|\mathbf{w}\|^2$; conversely Bounded Support Vectors are those inside the margin or those in the wrong class side. An interesting aspect of SVM is that SVM takes decisions on samples that are at the boundary of the two classes; this feature makes SVM powerful but, at the same time from the cognitive point of view, taking decisions on border samples could seem unjustified; practice shows that this is not the case and

that SVM is very effective in most classifications domains [59].

SVM has a direct geometric interpretation; the cost function hinge + regularization maximize the margin between the two (reduced) convex hulls of the data.

Support Vector Machine functional poses the problem of its optimization: standard optimization routines can be used such as the Interior Point method [58], but over the time very efficient ad-hoc optimizations routines were developed.

The probably mostly used routine for optimization is the Sequential Minimal Optimization algorithm [60] enriched by the successive modifications by [61] and [62]. This algorithm at every iteration first selects the pair of α_i with some heuristic then optimize them in closed form. This strategy allows for very low cost atomic iterations; a pair of α_i are used because one has to grant the linear constraint $\sum y_i \alpha_i = 0$ and so moving only one variable it could be impossible. The main motivation by which SMO is so effective is again due to the SVM loss function; this loss function, as said before induces sparsity on α_i this in turn means that a possibly big number of α_i does not contribute to f evaluation; thus the patterns that produce errors are only the Bounded Support Vectors, instead True Support Vectors are at the edge. For this reason during the optimization a restricted set of points violates the KKT conditions; this set is so restricted that a caching strategy of the kernel values is very important to speed-up computations. Hence SVM optimization carries an intrinsic redundancy and locality of the variables to be optimized; another technique that can be employed is the *shrinking* method which tries to indentify as soon as possible the non support vectors and thus eliminating them from the *active set*. The fact that SMO uses low computational resources derives from its capability of defining a pre-defined buffer, a cache, for kernel values, and then using only that amount of memory to compute and store on-the-fly the kernel values; instead in RLS one is constrained in computing all the kernel matrix and then performing optimization.

The main drawback of SMO is that, being iterative, its speed performance strongly depends on condition of the matrix K and on the regularization pa-

parameter C [61]. An high C value usually slows down SMO considerably; fortunately, by practice, one sees that the optimal C is usually less 10^3 so mitigating such a problem.

SMO has been extended for Kernel Logistic Regression [54] and inherits the SMO skill to be computationally not so demanding; conversely because the loss is different and there is no sparsity, performances are not so impressive [54]

An useful SVM variant that will be used through this thesis in two works is that in which the bias b is removed from the model. It is not difficult to show that the dual of this modified SVM is:

$$\begin{aligned} \min_{\alpha} \frac{1}{2} \alpha^t Q \alpha - \sum_{i=1}^n \alpha_i \\ 0 \leq \alpha_i \leq C \quad \forall i \end{aligned} \quad (2.103)$$

such dual has the advantage that there is no linear constraint, thus optimization can take place one α_i at each iteration; the removal of b is not dramatic because in the linear case one augment the original space X by a fixed feature of value 1 and in the non linear case b is essentially useless, intuitively in a infinite dimensional space sacrificing a shift is not so important; for a complete discussion about the role of b one can read the study in [63].

2.2.2.3 Kernel Logistic Regression

Kernel Logistic Regression is a powerful regression/classification tool. In addition to usual kernel machines properties such as non linearity and convexity it endows a probabilistic model; thus every prediction not only gives the class label but also the associated confidence. The probabilistic model underlying KRL is the logistic model. Given a model f , in the general case one has:

$$p = Pr(y|\mathbf{x}) = \frac{1}{1 + e^{-yf(\mathbf{x})}} \quad (2.104)$$

i.e. for the class +1:

$$p = Pr(y = 1|\mathbf{x}) = \frac{1}{1 + e^{-f(\mathbf{x})}} \quad (2.105)$$

The primal formulation of KLR [54] expressed in a SVM-like style as:

$$\begin{aligned} C \sum_{i=1}^n g(\xi_i) + \frac{\|\mathbf{w}\|^2}{2} \\ -y_i(\mathbf{w}\mathbf{x}_i) = \xi_i \quad \forall i \\ g(\xi_i) = \ln(1 + e^{\xi_i}) \quad \forall i \end{aligned} \tag{2.106}$$

As per SVM it is convenient to use a Lagrangian formulation.

$$\mathcal{L} = C \sum_{i=1}^n g(\xi_i) + \frac{\|\mathbf{w}\|^2}{2} + \sum_{i=1}^n \alpha_i [-\xi_i - y_i(\mathbf{w} \cdot \mathbf{x}_i)] \tag{2.107}$$

The corresponding KKT conditions are:

$$\nabla_{\mathbf{w}} \mathcal{L} = \mathbf{w} - \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i = 0 \tag{2.108}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = C \dot{g}(\xi_i) - \alpha_i = 0 \tag{2.109}$$

Now one defines the auxiliary function $G(\delta)$ where $\delta = \alpha/C$.

$$G(\delta) = \delta \xi_i - g(\xi_i) \tag{2.110}$$

Deriving G with respect to δ and using (2.109) one gets:

$$\frac{dG}{d\delta} = \delta \frac{d\xi_i}{d\delta} + \xi_i - \dot{g}(\xi_i) \frac{d\xi_i}{d\delta} = \xi_i = [\dot{g}]^{-1}(\delta) \tag{2.111}$$

thus $\dot{G}(\delta) = [\dot{g}]^{-1}(\delta)$, so $G(\delta)$ can be written using this relation. Recalling that $g(\xi) = \ln(1 + e^\xi)$ one gets that its derivative is $\dot{g}(\xi) = \frac{e^\xi}{1+e^\xi}$. Its inverse is $[\dot{g}]^{-1} = \ln(u/(1-u))$. From previous computations $\dot{G}(\delta) = [\dot{g}]^{-1}$ then one concludes that $\dot{G}(\delta) = \ln(\delta/(1-\delta))$.

Thus holds too:

$$G(\delta) = \delta \ln(\delta) + (1-\delta) \ln(1-\delta) \tag{2.112}$$

Hence the Lagrangian can be written as:

$$\mathcal{L} = C \sum_{i=1}^n g(\xi_i) + \frac{1}{C} (\alpha_i [-\xi_i - y_i(\mathbf{w} \cdot \mathbf{x}_i)]) + \frac{\|\mathbf{w}\|^2}{2} \tag{2.113}$$

Further:

$$\mathcal{L} = \frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^n g(\xi_i) - \frac{\alpha_i}{C} \xi_i - \sum_{i=1}^n \alpha_i y_i \mathbf{w} \mathbf{x}_i \tag{2.114}$$

It is not difficult to recognize that the term inside the first sum is the auxiliary function $-G(\delta)$. Using the first KKT and $G(\delta)$ one gets:

$$\mathcal{L} = \frac{\|\mathbf{w}\|^2}{2} - C \sum_{i=1}^n G\left(\frac{\alpha_i}{C}\right) - \|\mathbf{w}\|^2 \quad (2.115)$$

Then using again the first KKT condition and setting $q_{ij} = y_i y_j k_{ij}$:

$$\mathcal{L} = -\frac{1}{2} \boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha} - C \sum_{i=1}^n G\left(\frac{\alpha_i}{C}\right) \quad (2.116)$$

Recalling that one needs $\max \mathcal{L}$ one gets:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \frac{1}{2} \boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha} + C \sum_{i=1}^n G\left(\frac{\alpha_i}{C}\right) \\ G\left(\frac{\alpha_i}{C}\right) = \frac{\alpha_i}{C} \ln\left(\frac{\alpha_i}{C}\right) + \left(1 - \frac{\alpha_i}{C}\right) \ln\left(1 - \frac{\alpha_i}{C}\right) \end{aligned} \quad (2.117)$$

This cost function is convex but not linear, nor quadratic thus its optimization is not an easy task: the Iterated Weighted Least Squares method [58] is a possible algorithm; a low resource cost alternative is SMO for KLR [54] or a generic interior point method.

A further note regards a link between lagrange multipliers and probabilities; rewriting the primal cost as:

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^n -\ln(p_i) \quad (2.118)$$

deriving with respect to \mathbf{w} and using the first KKT one gets:

$$\mathbf{w} = C \sum_{i=1}^n (1 - p_i) y_i \mathbf{x}_i \quad (2.119)$$

At the optimum this equation shows an intimate relation between α_i and p_i :

$$\alpha_i = C(1 - p_i) \quad (2.120)$$

KLR and in particular this relation will be used when discussing the biased version of KLR.

2.2.2.4 Regularization and Learning

So far the statistical and function approximation point of view have been exposed. Classical and kernel regularization methods have been presented, however one can wonder why using $\|\mathbf{w}\|^2$ as regularizer instead of another one. This section links SVM to the d_{vc} concept explained in the previous sections and show that indeed SVM minimizes the d_{vc} so it is coherent with Statistical Learning Theory.

The concept d_{vc} is not the only linked to Regularization, others are: Maximal Discrepancy [26], Rademacher Complexity [28], Stability [29], Margin and PAC bounds [27]. Here an important result as representative of the link between learning (i.e. d_{vc}) and $\|w\|^2$ is reported: Vapnik and Chervonenkis showed [3] that for a class of classifiers (gap tollerant) that slightly differs from SVM the following holds:

Theorem 2.2.2. *Consider hyperplanes $\mathbf{w} \cdot \mathbf{x} = 0$ where $\min_i |\mathbf{w} \cdot \mathbf{x}_i| = 1$ for a set of points X . The set of decision functions such that $|\mathbf{w}| \leq A$ has VC dimension satisfying:*

$$d_{vc} \leq R^2 A^2 \quad (2.121)$$

where R is the radius of the smallest sphere around the origin containing X .

This result, despite not directly derived for SVM, suggests that minimizing $\|\mathbf{w}\|^2$, or equivalently maximizing the margin $2/\|\mathbf{w}\|^2$ leads to minimizing the d_{vc} of the classifier; thus machines that minimize a loss function and constrain $\|\mathbf{w}\|^2$ indeed perform Structural Risk Minimization.

One should observe that Structural Risk Minimization prescribes to minimize a loss function while at the mean time constraining the space of functions; with such view the learning process is as:

$$\begin{aligned} \min_{\mathbf{w}} \sum_{i=1}^n L(\mathbf{w}, X, \mathbf{y}) \\ \|\mathbf{w}\|^2 \leq \rho^2 \end{aligned} \quad (2.122)$$

In SRM the regularization is of Ivanov kind; indeed it is not clear at all if Ivanov regularization is completely equivalent to Tikhonov regularization; despite this SRM reasonably explains why SVM and kernel methods work.

Clearly L_2 regularization is not the only possible choice, others are entropy maximization and L_1 norms: the choice of L_2 is due to the fact that allows using RKHS and so the representer theorem; indeed such theorem does not hold for L_1 regularization such as for the Lasso classifier [64]. Recently using a PAC argument [65] it has been shown that also a class of learning algorithm regularized by Shannon entropy on $\|\mathbf{w}\|_1$ still grants generalization. The fact that both L_2 and Shannon entropy allows effective learning is quite surprising and seems to suggest that something more general holds; in particular the following conjecture can be stated.

Conjecture 2.2.1. *Algorithms of the form:*

$$\min_{\mathbf{w}} \sum_{i=1}^n L(\mathbf{w}, X, \mathbf{y}) + \lambda \mathcal{R}_\alpha(\mathbf{w}) \quad (2.123)$$

allows learning, where learning means that the negative Renyi entropy $\mathcal{R}_\alpha(\mathbf{w}) = -\frac{1}{1-\alpha} \log \left(\sum_{j=1}^d |w_j|^\alpha \right)$ for certain values of α allows a generalization error bound where an increasing function of \mathcal{R}_α is the complexity term.

Such a result would generalize the link between learning and regularization methods. In particular for two different values of α , namely $\alpha = 1$ and $\alpha = 2$ the result is already proved in [65] and in [3] respectively; however is not proved in the general form. In such class of algorithms one can distinguish some cases:

- For $\alpha = 1$ one gets $\mathcal{R}_1 = \sum_{i=1}^d |w_i| \log |w_i|$; thus one gets maximization of Shannon entropy as regularizer. This regularizer allows learning [65].
- For $\alpha = 2$ one gets $\mathcal{R}_2 = \log \sum_{i=1}^d |w_i|^2 = \log \|\mathbf{w}\|^2$ that up to the monotone mapping given by the logarithm one gets $\|\mathbf{w}\|^2$; thus one gets Kernel Machines. This regularizer allows learning [3].
- For $\alpha = 2$ one gets $\mathcal{R}_{2b} = \log \sum_{i=1}^d |w_i|^2 < \sum_{i=1}^d \log |w_i|^2$ that up to the monotone mapping given by the logarithmic one gets $\|\mathbf{w}\|_1$; thus one gets Lasso-type [64] machines. It is not clear if this regularizer allows learning however it is used in features selection problems.

- For $2 < \alpha < \infty, 1 < \alpha < 2, \alpha < 1$ nobody has studied these intermediate cases.
- For $\alpha = \infty$ one gets $\mathcal{R}_\infty = \log \sup_{i=1,\dots,d} |w_i|$. Nobody has studied this case

2.2.3 Recent Neural Models

This section gives a brief overview on learning tools that will be studied, used and enhanced through this thesis.

2.2.3.1 Circular Back Propagation Networks for Classification

The Circular Back Propagation network model (CBP) [66] extends the classical Multilayer Perceptron (MLP) [66]. An auxiliary term, whose value is the Euclidean norm of the input vector, \mathbf{x} , extends the input space, R^m , in the CBP model (see figure 2.2). The transfer function of the j -th neuron in the input CBP layer is:

$$\sigma_j(\mathbf{x}) = \sigma \left(\sum_i^m w_{i,j}^{(I)} x_i + w_{0,j}^{(I)} + w_{m+1,j}^{(I)} \sum_i^m x_i^2 \right); \quad j = 1, \dots, n_h \quad (2.124)$$

where n_h is the number of neurons of the upper layer. In a typical three-layer CBP network, the classification function is given by the typical MLP formula:

$$f_{CBP}(\mathbf{x}) = \sum_j^{n_h} w_j^{(II)} \sigma_j(\mathbf{x}) + w_0^{(II)} \quad (2.125)$$

The set of real coefficients at each level $\{\mathbf{w}^{(l)}, l = I, II\}$ are the model parameters that are adjusted by the training procedure.

The CBP model exhibits several significant advantages over several classical adaptive classifiers: the most significant regards the fact that this network shows a remarkable representation ability without penalizing the model complexity significantly. At the same time, it has been proved that the CBP approach compares favorably with popular architectures such as radial basis functions networks [66]. CBP classifiers have been tested and applied successfully in a wide variety of practical applications [66].

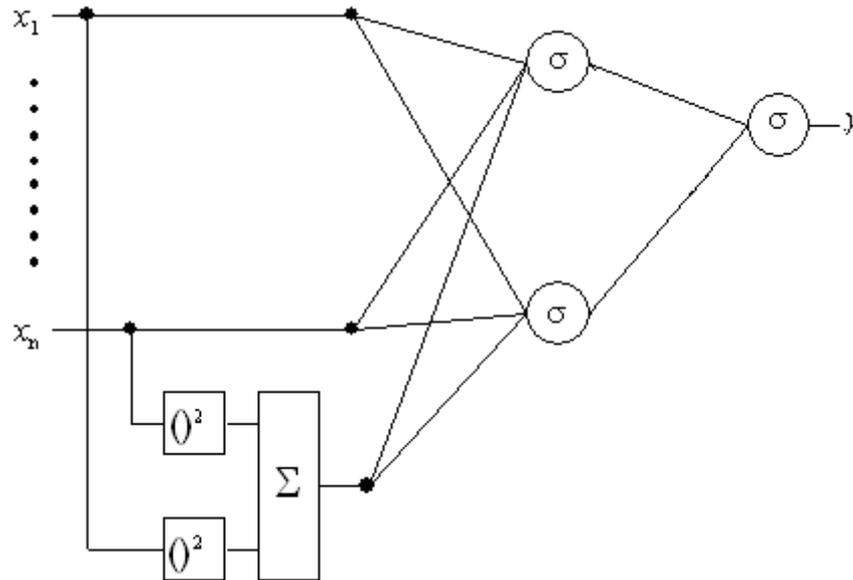


Figure 2.2: Circular Back Propagation Network with the circular input evidenced

2.2.3.2 Random Neural Networks

Random Neural Networks are one hidden layer neural networks where the hidden weights are randomly set. The idea of building a circuitual network with random connections is ancient and dates back to a Turing note [67] (1948) called *Intelligent Machinery* published only in 1968 where he first had the intuition that hierarchical structures (digital circuits) randomly connected could endow learning skills; Turing called this kind of networks as A-type networks. Random Neural Networks, to some extent, are analogous to A-type networks because in their structures there is a consistent randomly generated portion; random neural networks were introduced by [68], further studied in [69] and recently again discovered and studied under the name of *Extreme Learning Machine* [70]; this name is due to the high speed of the learning process; from now referring to ELM means referring to random neural networks. It has been a debate [71] whatever ELM is different from original random neural networks proposed in [68], [69] or not; this discussion is out of the scope of this work, the aim here is improving on the baseline random neural network model. In such networks the connection parameters

between the input and the hidden layer of a feed-forward neural network are preset randomly and are not subject to optimization. Reducing the training process to the adjustment of the output layer yields a convex optimization problem that is solved efficiently by a linear system.

Training algorithms such as Back-Propagation (BP) [36] adjust the weights of multi-layer networks by means of a gradient-descent process. The Extreme Learning Machine (ELM) approach [70] offers promising perspectives to overcome the typical issues in BP-based implementations, namely, possibly slow convergence rates, the critical tuning of optimization parameters [72], and the presence of local minima that call for multistart and re-training strategies.

The learning problem setting requires a training set, \mathbf{X} , of n labeled pairs (\mathbf{x}_i, y_i) , where $\mathbf{x}_i \in \mathbf{R}^m$ is the i -th input vector and $y_i \in \mathbf{R}$ is the associate expected 'target' value. A single-layer feedforward network connects the input layer (having m neurons) to the "hidden" layer (having N_h neurons) through a set of weights $\{\hat{w}_j \in \mathbf{R}^m; j=1, \dots, N_h\}$. The j -th hidden neuron embeds a bias term, \hat{b}_j , and a nonlinear "activation" function, $g(\cdot)$; the neuron response to an input stimulus, \mathbf{x} , is:

$$a_j(\mathbf{x}) = g(\hat{w}_j \cdot \mathbf{x} + \hat{b}_j) \quad (2.126)$$

Note that (2.126) can be further generalized to a wider class of functions [73] but for the subsequent analysis this aspect is not relevant.

A vector of weighted links, $\bar{w}_j \in \mathbf{R}^{N_h}$, connects the hidden layer to the output neuron, having bias \bar{b} . As a result, the overall output function, $f(\mathbf{x})$, of the single-layer neural network is written as:

$$f(\mathbf{x}) = \sum_{j=1}^{N_h} \bar{w}_j a_j(\mathbf{x}) + \bar{b} \quad (2.127)$$

It is convenient to define an "activation matrix", \mathbf{H} , such that the entry $\{h_{ij} \in \mathbf{H}; i=1, \dots, n; j=1, \dots, N_h\}$ is the activation value of the j -th hidden neuron for the i -

th input pattern. The \mathbf{H} matrix is:

$$\mathbf{H} \equiv \begin{pmatrix} g(\hat{\mathbf{w}}_1 \cdot \mathbf{x}_1 + \hat{b}_1) & \cdots & g(\hat{\mathbf{w}}_{N_h} \cdot \mathbf{x}_1 + \hat{b}_{N_h}) \\ \vdots & \vdots & \vdots \\ g(\hat{\mathbf{w}}_1 \cdot \mathbf{x}_N + \hat{b}_1) & \cdots & g(\hat{\mathbf{w}}_{N_h} \cdot \mathbf{x}_N + \hat{b}_{N_h}) \end{pmatrix} \quad (2.128)$$

In the ELM model, the terms \hat{w}_j and \hat{b}_j in (2.126) are set randomly and are not subject to any adjustment. Since the quantities \bar{w}_j and \bar{b} are the only degrees of freedom of the ELM learning process, the training problem reduces to the minimization of the convex cost:

$$\min_{\{\bar{\mathbf{w}}, \bar{b}\}} \|\mathbf{H} \bar{\mathbf{w}} - \mathbf{y}\|^2 \quad (2.129)$$

Thus a pseudo-inversion operation yields the unique L_2 solution:

$$\bar{\mathbf{w}} = \mathbf{H}^+ \mathbf{y} \quad (2.130)$$

The simple, efficient procedure to train an ELM consists in the following steps:

1. Randomly set the input weights \hat{w}_i and bias \hat{b}_i for each hidden neuron;
2. Compute the activation matrix, \mathbf{H} , as per (2.128);
3. Compute the output weights by solving a pseudo-inverse problem as per (2.130).

In spite of the apparent simplicity of the ELM approach, the crucial result is that even random weights in the hidden layer endow a network with a satisfactory representation ability. The theory derived in [70] proves that the ELM with $N_h = n$ hidden nodes can approximate any function.

2.3 Unsupervised Learning, KWM and Random Projections

This section introduces some unsupervised learning tools that will be used throughout this thesis, these are: kernel k-means [74], spectral clustering [75] and plastic neural gas [76]. Moreover Random projections and the K-Winner Machine model are briefly introduced [77].

2.3.1 K-Means and Kernel K-Means

The conventional k-means paradigm supports an unsupervised grouping process, [78] which partitions the set of n samples into a set of n_c clusters, $C_j (j = 1, \dots, n_c)$. In practice, one defines a “membership vector” $\mathbf{m} \in \{1, \dots, n_c\}$ of length n , which indexes the partitioning of input patterns over the n_c clusters as: $m_i = j \Leftrightarrow \mathbf{x}_i \in C_j$, otherwise $m_i = 0; i = 1, \dots, n$. It is also useful to define a “membership function” $\delta_{ij}(\mathbf{x}_i C_j)$, that defines the membership of the i -th sample to the j -th cluster: $\delta_{ij} = 1$ if $m_i = j$, and 0 otherwise. The result of the clustering strategy is twofold: the method assigns the input samples to any of the n_c clusters uniquely, and, therefore, the cluster centroids (often also called prototypes) can be computed explicitly, and are the average positions of the samples within the respective clusters. With the above definitions, the number of members of a cluster is expressed as

$$n_j = \sum_{i=1}^n \delta_{ij} \quad j = 1, \dots, n_c \quad (2.131)$$

and the cluster centroid is given by:

$$\mathbf{w}_j = \frac{1}{n_j} \sum_{i=1}^n \mathbf{x}_i \delta_{ij} \quad j = 1, \dots, n_c \quad (2.132)$$

The k-means algorithm tries to minimize the average distortion cost:

$$L(X, \mathbf{w}) = \frac{1}{n} \sum_{j=1}^{n_c} \sum_{i \in C_j}^{n_j} \|\mathbf{x}_i - \mathbf{w}_j\|^2 \quad (2.133)$$

K-Means algorithm performs this minimization with a simple iterative scheme:

Algorithm 1 K-means clustering**Require:** A training set of n data, \mathbf{x}_i **Ensure:** Clusters membership \mathbf{m}

- 1: Initialize \mathbf{m} with random memberships $m_i \in \{0, 1, \dots, n_c\}$; mark \mathbf{m} as ‘modified’
- 2: **while** \mathbf{m} is modified **do**
- 3: Compute $d(\mathbf{w}_j, \mathbf{x}_i)^2, j = 0, 1, \dots, n_c; i = 1, \dots, n$
- 4: $m_i = \arg \min_j d(\mathbf{w}_j, \mathbf{x}_i)^2$
- 5: **end while**

This algorithm, except in pathological situations, converges very fast to sub-optimum of the distortion cost.

A nice property of k-means is that only the distance function is needed to perform the iterations, and there is no need to explicitly know the centers \mathbf{w}_j . This fact suggests that one can generalize the algorithm to its kernel version: the key idea underlying kernel-based k-means [74] clustering is indeed that the actual coordinates of the cluster centroids may *not* be known explicitly, as long as one is just interested in the memberships of samples to the various groups. Under such assumption, one can include the kernel-based approach into the k-means formulation as follows.

First, one assumes that a function, ϕ , can map any element, \mathbf{x} , of the input space into a corresponding position, $\phi(\mathbf{x})$, in a Hilbert space. The mapping function defines the actual ‘Kernel’, which is formulated as the expression to compute the inner product:

$$K_{uv} = \phi(\mathbf{x}_u) \cdot \phi(\mathbf{x}_v) \quad (2.134)$$

The Hilbert space spanning vectors ϕ , as already discussed, can have an arbitrary dimension (even infinite) and just requires that an inner product be defined.

The kernel-based version of the k-means algorithm replicates the basic partitioning schema of the baseline k-means in the Hilbert space, where the cen-

centroid positions are given by the averages of the mapping images, ϕ_u :

$$\phi(\mathbf{w}_j) = \frac{1}{n_j} \sum_{i=1}^n \phi_i \delta_{ij} \quad j = 1, \dots, n_c \quad (2.135)$$

The ultimate result of the clustering process is the membership vector, \mathbf{m} , which determines prototype positions even though they cannot be stated explicitly. As a consequence, for a sample, \mathbf{x}_i , the distance in the Hilbert space from the mapped image, ϕ_u , to the cluster $\Psi_j = \phi(\mathbf{w}_j)$ can be worked out as:

$$\begin{aligned} d(\phi_i, \Psi_j)^2 &= \left\| \phi_i - \frac{1}{n_j} \sum_{k=1}^n \phi_k \right\|^2 = \\ &= \left(\frac{1}{n_j} \sum_{k=1}^n \phi_k \delta_{kj} \right) \cdot \left(\frac{1}{n_j} \sum_{k=1}^n \phi_k \delta_{kj} \right) + \phi_i \cdot \phi_i - \frac{2}{n_j} \sum_{k=1}^n \delta_{kj} (\phi_i \cdot \phi_k) = \\ &= \frac{1}{(n_j)^2} \sum_{k,l=1}^n \delta_{kj} \delta_{lj} K_{kl} + K_{ii} - \frac{2}{n_j} \sum_{k=1}^n \delta_{kj} K_{ik} = \\ &= K_{ii} + \frac{1}{(n_j)^2} \sum_{k,l=1}^n \delta_{kj} \delta_{lj} K_{kl} - \frac{2}{n_j} \sum_{k=1}^n \delta_{kj} K_{ik} \end{aligned} \quad (2.136)$$

By using the last expression, which includes only kernel computations, one can identify the closest prototype to the image of each input pattern, and assign sample memberships accordingly. The overall feature-space k -means algorithm can be outlined as follows:

Algorithm 2 The feature-space version of kernel k -means clustering

Require: A training set of n data, \mathbf{x}_i

Ensure: Clusters membership \mathbf{m}

- 1: Initialize \mathbf{m} with random memberships $m_i \in \{0, 1, \dots, n_c\}$; mark \mathbf{m} as 'modified'
 - 2: **while** \mathbf{m} is modified **do**
 - 3: Compute $d(\Psi_j, \phi_i)^2, j = 0, 1, \dots, n_c; i = 1, \dots, n$
 - 4: $m_i = \arg \min_j d(\Psi_j, \phi_i)^2$
 - 5: **end while**
-

The crucial advantage in moving to a Hilbert space is that a representation that might be contorted in the original space may turn out to be straightforward in the mapping space. Thus, even though individual points are no longer identifiable explicitly, one is interested in structures among points.

2.3.2 Plastic Neural Gas

Plastic Neural Gas (PGAS) [76] extends the Neural Gas (NGAS) [76] model of Vector Quantization, and uses an iterative process to perform Vector Quantization (VQ). The training algorithm processes the representative vectors of n input patterns $\{\mathbf{x} \in R^m\}$ and positions a ‘codebook’ of n_c prototypes in the data space. If one denotes the j -th cluster as C_j and indicates with n_j the number of patterns lying in C_j , PGAS minimizes the total distortion cost (2.133). The optimization process is necessarily sub-optimal because (2.133) implies a problem of non-polynomial complexity. PGAS uses a strategy analogous to simulated annealing [76] to escape from local minima. The PGAS training strategy offers two significant advantages: first, it can adaptively set the correct number of VQ prototypes at run time [76]; secondly, it prevents the occurrence of ‘dead vectors’, i.e., void prototypes that cover empty partitions of the data space.

To use PGAS as a classifier building model, a calibration process [76] completes the unsupervised training process and labels the tessellation of the data space induced by the positions of the prototypes. Each partition/prototype is labeled according to the predominant class; from a cognitive viewpoint, the latter step aims to reproduce the conditional distribution of classes. After training and calibration, new samples are classified according to the class of the nearest prototype.

2.3.3 Spectral Clustering

Spectral Clustering is a recently proposed technique to perform clustering in a possibly non linear setting. The name of *Spectral Clustering* SC is quite misleading because it is not a clustering algorithm but a pre-processing step followed by a k-means clustering step; thus SC is more technique to pre-process data. The attribute *Spectral* derives from the fact the embedding of the data is get by using the spectrum of the Laplacian of a proper similarity matrix.

Indeed there are several variants of spectral clustering; here it is presented the normalized spectral clustering [75]. A typically used similarity metrix is

that given by the usual Gaussian kernel \mathbf{K} . The normalized graph Laplacian [75] is an approximation of the Laplace-Beltrami operator on a discrete setting; it is defined as:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{K} \mathbf{D}^{-1/2} \quad (2.137)$$

where \mathbf{D} is a diagonal matrix with:

$$D_{ii} = \sum_{j=1}^n K_{ij} \quad (2.138)$$

It is known that if in the data there k clusters, then \mathbf{L} has k zero eigenvalues, that are also the smallest due to positive definiteness of the Laplacian; clearly in real problems one check for small eigenvalues, null are impossible. Roughly speaking one recover the embedding of the data by the spectral properties of the Laplacian, and then, in that space performs clustering. In pseudocode terms one has:

Algorithm 3 Spectral Clustering

Require: A training set of n data, \mathbf{x}_i

Ensure: Clusters membership \mathbf{m}

- 1: Build a positive definite similarity matrix \mathbf{K}
 - 2: Build the normalized Laplacian according to 2.137
 - 3: Compute the k eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_k$ associated to the k smallest eigenvalues
 - 4: Form the matrix $\mathbf{U} \in \mathbb{R}^{n \times k}$ that contains the eigenvectors as columns
 - 5: Normalize \mathbf{U} row-wise
 - 6: Perform clustering (e.g. k-means) on \mathbf{U} and return the membership vector \mathbf{m}
-

At first glance it is not clear why such algorithm should work: SC can be explained from different points of views, mainly from a graph cut and from a random walk point.

Defining the correct number of clusters is an open problem in clustering theory; an effective rule of thumb in spectral clustering is using the eigengap between eigenvalues $\lambda_1, \dots, \lambda_j$; whenever the difference $|\lambda_i - \lambda_{i+1}|$ is more than a given threshold then one defines the number of clusters as $k = i + 1$ and stop

computing eigenvectors.

Intuitively it seems that Spectral Clustering can be related to kernel k-means. In SC one build a Gaussian kernel, build the normalized Laplacian and the perform k-means on the embedding; in KK-Means one builds the Gaussian kernel, and then on that space perform clustering. It is intuitive the fact that SC with respecto to KK-Means performs one more step that is the Laplacian embedding however the algorithms look similar. It has been shown [79], indeed, that a form of weighted Kernel K-Means is equivalent to spectral clustering; interestingly weighted Kernel K-Means does not use any (computationally demanding and not always stable) eigenvalue decomposition, thus one could use weighted kernel k-means to emulate the behaviour of Spectral Clustering; this is important because Spectral Clustering results are usually significantly better than that of Kernel K-Means but at the higher cost of computing eigenvectors. Here by weight one means a weight associated to each pattern when computing the loss; more explicitly, given weights w_i the cost minimized by a weighted kernel k-means is:

$$L(X, \mathbf{w}) = \frac{1}{n} \sum_{j=1}^{n_c} \sum_{i \in C_j}^{n_j} w(\mathbf{x}_i) \|\mathbf{x}_i - \mathbf{w}_j\|^2 \quad (2.139)$$

and the clusters centers now are:

$$\Psi_j = \frac{\sum_{i \in C_j} w(\mathbf{x}_i) \phi(\mathbf{x}_i)}{\sum_{i \in C_j} w(\mathbf{x}_i)} \quad (2.140)$$

Now the distance $d(\Psi_j, \phi(\mathbf{x}_i))^2$ is given by:

$$K_{ii} + \frac{\sum_{k,l \in C_j} w(\mathbf{x}_k) w(\mathbf{x}_l) K_{kl}}{(\sum_{k \in C_j} w(\mathbf{x}_k))^2} - \frac{2 \sum_{k \in C_j} w(\mathbf{x}_k) K_{ik}}{\sum_{k \in C_j} w(\mathbf{x}_k)} \quad (2.141)$$

In particular it is shown that using as weights the diagonal of \mathbf{D} and considering the kernel $\mathbf{K} = \mathbf{D}^{-1} \mathbf{K}_g \mathbf{D}^{-1}$ where \mathbf{K}_g is the original unweighted kernel of kernel k-means then, weighted kernel k-means minimizes the same cost of the normalized cut problem, that it is one of the formulations of spectral clustering.

2.3.4 Random Projection

Random Projections represent a recent, increasingly popular approach to support feature reduction in a simple and efficient way. These methods are relatively inexpensive (due to the intrinsic parallelism) from a computational viewpoint and yield reliable results on complex domains [80]; in the present context, they are used as a data-analysis tool to support visual inspection of the observed domain.

The RP formalism stems from the following fundamental result on manifolds [80]:

Lemma 2.3.1. Johnson-Lindenstrauss lemma (JL-Lemma) [80] For any $0 < \varepsilon < 1$ and any integer t , let k be a positive integer such that:

$$k \geq 4(\varepsilon^2/2 - \varepsilon^3/3)^{-1} \ln(t) \quad (2.142)$$

Then with probability $O(1/t^2)$ for any set X of points in R^m , there is a map $f : R^m \rightarrow R^k$ such that for all $\mathbf{x}, \mathbf{z} \in X$

$$(1 - \varepsilon) \leq \frac{\|f(\mathbf{x}) - f(\mathbf{z})\|^2}{\|\mathbf{x} - \mathbf{z}\|^2} \leq (1 + \varepsilon) \quad (2.143)$$

The function $f()$ is rendered by projecting the original data X into the random subspace spanned by a random matrix R . This matrix is made up of d rows, k columns, every matrix entry is distributed as $N(0, 1)$ and columns have unitary length.

The JL-Lemma ensures that, when comparing the distance values between a pair of patterns in the original space $\{\mathbf{x}, \mathbf{z}\}$ and in the mapped space $\{f(\mathbf{x}), f(\mathbf{z})\}$, the distortion in distances is less than $\varepsilon \cdot \|\mathbf{x} - \mathbf{z}\|^2$. To address the projected space, \hat{X} , one should just build up a random matrix, R , (having size $d \times n_r$) as per JL-Lemma when $k = n_r$. Then the projected space is obtained from the original space by:

$$\hat{X} = \frac{1}{\sqrt{n_r}} \mathbf{X}R \quad (2.144)$$

The distance-preserving property makes RP methods quite interesting for domain inspection. The mapping function, f , is the operational core of the

overall method; if one imposes that the target space is lower-dimensional than the original one, ($k \ll m$) the overall framework can be regarded as a dimensionality-reduction compression process. This technique has been successfully used in text mining domains [81] for clustering purposes. In the current research, the projection method will be both for feature reduction in text mining problems and as a visual inspection tool by imposing $k = 2$.

2.3.5 KWM Classifiers and Prediction Error Estimation

K-Winner machine [77] is a classification algorithm whose Structural Risk Minimization properties can be defined analytically. The training strategy of the K-Winner Machine (KWM) model first develops a representation of the data distribution by means of an unsupervised process, i.e. clustering, then applies a calibration process to train a supervised classifier. A detailed outline of the KWM training algorithm is given in [77]. At run time, each point in the data space is classified locally, under the cognitive assumption that the risk in the classification outcome for a given point decreases when more and more neighboring prototypes concur in the classification of that point. As opposed to conventional ensemble methods, a KWM requires a complete agreement among the set of best-matching prototypes and does not involve any majority counting; the smallest set will include the nearest prototype only.

The advantage of applying Statistical Learning Theory to the KWM model mainly lies in the computation of generalization bounds at the local level. The analysis presented in [77] adopted the formulation based on the Vapnik-Chervonenkis dimension, and derived several analytical properties of KWMs, including the Vapnik-Chervonenkis dimension and the analytical expression of the Growth Function of the family of classifiers used in the KWM model.

The resulting theory proves that one can compute an error bound, $R[f]$, for each agreement level, k , and more importantly, that such a bound is a non-increasing function when k increases. This confirms the intuitive notion that the risk in a classification decision about a given point should be reduced by the concurrence of several neighboring prototypes.

A crucial feature of the KWM model is that, by using the prototype-agreement

criterion at run time, any point in the data space is characterized by a local bound to the classification error. Such a bound, that is the instantiation of Vapnik bound 2.13 for the KWM case, has been derived analytically [77]: the main result states that with probability $1 - \delta$ and indicating with n_h the number of prototypes, it holds:

$$\chi(k) = \frac{4}{n} \left(\left\lfloor \frac{n_h}{k} \right\rfloor \ln 2 - \ln \frac{\delta}{4} \right) \quad (2.145)$$

$$R^k[f] \leq R_{emp}^k[f] + \frac{\chi(k)}{2} \left(1 + \sqrt{1 + \frac{4R_{emp}^k[f]}{\chi(k)}} \right) \quad (2.146)$$

As a consequence, unsupervised prototype positioning sharply reduces the bounding term in (2.146). By contrast, the KWM training algorithm does not provide any a-priori control over the empirical training error, due to the unsupervised training mechanism. This brings about the problem of model selection, which is usually tackled by a tradeoff between accuracy (classification error in training) and complexity (number of prototypes).

2.4 Model Selection

Given the previous notions of what is the learning problem, what are the tools to cope with it the fundamental open problem is how to select the proper equilibrium between memory (loss function) and abstraction (regularization). In general one can say that when one has plenty of data a model selection is purely a computational problem; one can use methods such as k-fold cross validation, leave-one-out, test set method on generalization bound and they are all valuable tools from the accuracy of their predictions; problems in this case can only arise from the very large scale of problems.

Opposite to this situation there is the possibility that the problem is a really small sample problem and methods based on an hold-out procedure cannot be practically or reliably applied; in this case using all the available patterns as training patterns is of paramount importance; to this aim generalization error bounds, provided that are tight enough, can be powerful tools.

Now a brief overview of hold-out and generalization bounds model selection methods is given:

2.4.1 Test Set method

In this method the available data is split in two disjoint sets; the training set and the test set. The training set is used to train the machine and the test is used to assess the accuracy and select the regularization and/or kernel parameters. Such a method makes impossible to use all the available data for training because of the split procedure; so one loses samples with this procedure. As already mentioned in large scale problems this procedure can be already reliable.

2.4.2 K-fold cross validation

This procedure is the generalization of the previous one. In this case one splits data in k-groups: perform training on one of the groups and test on the others; then iterates this procedure for all the k folds and the average error rate is taken as estimation of the generalization error. This procedure is reliable, however loses more data than the previous one. Moreover, given k

different training results, *what is the model to be used when operating on new data?* There is no clear answer, one should pick a good value of the regularization/kernel parameter and than use it on the whole dataset; this is an intuitive answer but is not rigorous.

2.4.3 Leave-One-Out

This procedure is the extremization of the previous; all the possible groups where only one test point is left out, are built and than averaged. This procedure is computationally expensive (except in particular cases such as for RLS [82]) and suffer from the same problems of the previous method.

2.4.4 Bootstrap

The bootstrap method [83] is an effective technique for model selection: the training set is built by extracting n patterns *with replacement* (duplicates can exist) from the original training set. The left out patterns can be used as an independent test set. Bootstrap theory shows that, on average, $n/e \sim 0.368n$ patterns are left for test set. The training set with duplicates is called *bootstrap replicate* and up to $N_b = \binom{2n-1}{n}$ different replicates can be generated. Usually $N_b = 1e3$ are sufficient for effective error estimation. The estimate of the generalization error is the average of the errors on the test sets.

2.4.5 Generalization error bounds

A technique frequently used in small sample problems [84][8] is given by generalization error bounds. Various methods, as MD, Rademacher and PAC-Bayesian methods are possible. All these bounds have been successfully employed for model selection [31][84][8] and are among the most tight for assessing the value of generalization error. Despite this fact, the gap between true generalization error and its corresponding bound is quite high: this is the main motivation which inspired most of the theoretical part of this thesis; in particular the need of structuring the hypothesis space in SRM lead to the study of two generalization of kernel machines: Tikhonov regularization with

a generalized regularizer and biased regularization. Both these introduced and studied generalizations allow to enforce a structure on the hypothesis space that SRM by itself does not guarantee.

3

Structuring the hypothesis space

This chapter collects what can be considered the theoretical contributions of this thesis. As first work the regularized mean problem is studied: this problem is analogous to Tikhonov regularization when fitting a constant function, i.e. the mean. In this controlled context the notion of oracular regularization is introduced and neural networks are used to predict the regularization parameter and thus the mean value. The extension of this work is given by approaching Tikhonov regularization from the oracular point of view; the main finding is that a more structured hypothesis space is needed to grant a certain notion of optimality; the regularizer $\|\mathbf{w}\|$ is substituted by the general form $\|\mathbf{T}\mathbf{w}\|$ and allows nice connections with the Vapnik concept of Universum [7].

Another attempt of structuring the hypothesis space \mathcal{F} is given by studying kernel machines based on biased regularization; first it is shown that MD for SVM can be considerably shrink by using a Ivanov-like biased regularization term, then using Tikhonov-like regularization terms allows to define bRLS, bSVM that are the biased versions of RLS and SVM respectively; these models, although general, are used to efficiently address the semi-supervised learning problem.

The last section of this chapter constitutes a simple proof on how to obtain an explicit generalization error bound when dealing with transductive learning and can be considered a completion and adaptation of a Vapnik result. The contributions of this chapter are in [4],[5],[6], [8],[10],[11].

3.1 The Regularized Mean Problem

Let us consider m samples $x_i \sim N(\mu, \sigma^2)$ as the elements of a set X . These samples are used to define the functional:

$$\mathfrak{S}(\xi; X, \alpha) \equiv \sum_{i=1}^m (x_i - \xi)^2 + \alpha \xi^2 \quad (3.1)$$

where $\alpha \geq 0$ is the regularization parameter. Functional (3.1) can be considered a generalization of its non regularized counterpart. When $\alpha = 0$ the minimum of $\mathfrak{S}(\xi; X, 0)$ with respect to ξ leads to the usual sample mean estimation. Differently, when $\alpha > 0$ a regularized mean value is obtained. Define the regularized mean value \bar{x} for given (x_1, x_2, \dots, x_m) and α as:

$$\bar{x}(\alpha) \equiv \arg \min_{\xi} \mathfrak{S}(\xi; X, \alpha) \quad (3.2)$$

Searching for the minimum of (3.1) leads to:

$$\bar{x}(\alpha) = \frac{\sum_{i=1}^m x_i}{m + \alpha} \quad (3.3)$$

the requisite for which $\bar{x}(\alpha)$ corresponds to the minimum of (3.1) is satisfied if:

$$\left. \frac{d^2 \mathfrak{S}}{d\xi^2} \right|_{\xi=\bar{x}} > 0 \Rightarrow \alpha > -m \quad (3.4)$$

The non regularized solution is for $\alpha = 0$; as anticipated, this is the classical sample mean estimator:

$$\bar{x}_0 \equiv \bar{x}(0) = \frac{\sum_{i=1}^m x_i}{m} \quad (3.5)$$

Equation (3.3) can be conveniently rewritten as:

$$\bar{x} = \lambda \bar{x}_0 \quad (3.6)$$

where:

$$\lambda \equiv \frac{m}{m + \alpha} \quad (3.7)$$

The term λ is useful to evidence the analogies of the regularized mean problem with the influential theory of James-Stein [85], as it will be shown later. The regularized mean, being an estimator, is characterized as usual by bias and variance. The bias term is simply given by:

$$b_{\bar{x}} \equiv \mathbf{E}_X \{\bar{x}\} - \mu = \lambda \mathbf{E}_X \{\bar{x}_0\} - \mu = (\lambda - 1)\mu \quad (3.8)$$

Lemma 3.1.1. *The variance $\sigma_{\bar{x}}^2 \equiv \mathbf{E}_X \{(\bar{x} - \mathbf{E}_X \{\bar{x}\})^2\}$ is :*

$$\sigma_{\bar{x}}^2 = \frac{\lambda^2 \sigma^2}{m} \quad (3.9)$$

Proof. See Appendix

A criterion to define an optimal value for λ can now be formulated. A possible functional, according to [85], measuring the closeness of the \bar{x} estimate to the expected value μ is:

$$L(\lambda; X, \mu) \equiv \mathbf{E}_X \{(\bar{x} - \mu)^2\} \quad (3.10)$$

After simple computations it can be shown that:

$$L(\lambda; X, \mu) = \sigma_{\bar{x}}^2 + b_{\bar{x}}^2 = \frac{\lambda^2 \sigma^2}{m} + (\lambda - 1)^2 \mu^2 \quad (3.11)$$

The value of λ minimizing (3.10) is the *oracular* value:

$$\lambda^{orac} = \arg \min_{\lambda} L(\lambda; X, \mu) = \frac{\mu^2}{\mu^2 + \frac{\sigma^2}{m}} \quad (3.12)$$

This value could be evaluated only after μ itself is known, which justifies the term “*oracular*”. Fig. 3.1 shows an example of the advantage gained estimating μ as $\bar{x} = \lambda^{orac} \bar{x}_0$ instead of the unregularized value $\bar{x} = \bar{x}_0$. On the horizontal axis μ ranges in the interval $[-1, +1]$; on the vertical axis, the gain obtainable by oracular regularization, $y = \mathbf{E}_X \{(\bar{x}_0 - \mu)^2\} - \mathbf{E}_X \{(\lambda^{orac} \bar{x}_0 - \mu)^2\}$ is reported using $\sigma^2 = 1$, $m = 10$.

A key quantity in the regularized mean problem is the number d_f of *degrees of freedom*.

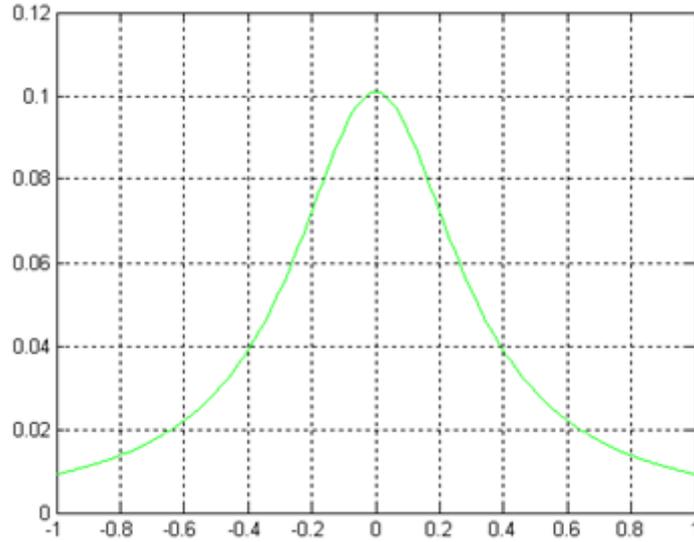


Figure 3.1: Advantage of oracular regularized solutions against non regularized: x axis is μ range, y axis is the quality metric

Lemma 3.1.2. *The number of degrees of freedom df of the regularized mean problem is:*

$$d_f(\lambda) = (\lambda - 1)^2(m\mu^2/\sigma^2 + 1) + (m - 1) \quad (3.13)$$

Proof. See Appendix

Taking $\lambda = \lambda^{orac}$ the above expression, after some manipulations, yields:

$$d_f(\lambda^{orac}) = m - \lambda^{orac} \quad (3.14)$$

This result evidences a direct connection between the oracular regularizer and its corresponding number of degree of freedom. It should be also stressed that, accordingly to [86], $d_f(\lambda) > m - 1$; the lowest d_f value is $m - 1$, corresponding to $\lambda = 1$, as for the usual variance unbiased estimator.

In order to obtain estimates of λ^{orac} , various statistical methods can be considered. Before doing this the connections between regularization and Stein [85] theory are developed.

3.1.1 Links with James-Stein theory

Stein's theory [85] deals with the so called n -means problem that can be outlined as follows:

1. Define n unknown parameters, the means vector μ : correspondingly define n Gaussian probability density functions that share the same variance σ^2
2. Pick one sample for each p.d.f. and denote by X this samples vector.

With these specifications, estimate the μ vector as:

$$\bar{X} = \arg \min_{\xi} \mathbf{E}_X \{ \|\xi - \mu\|_2^2 \} \quad (3.15)$$

In [85] it is shown that for $n \geq 3$ the estimator $\bar{X} = X$ is “inadmissible”, that is $\bar{X} = X$ is not the best possible estimator of μ for the loss that appears at right hand side in (3.15).

In [85] it is also proved that for $n \geq 3$ the best estimator for the vector μ is:

$$\bar{X} = \left(1 - \frac{(n-2)\sigma^2}{\|X\|^2} \right) X \quad (3.16)$$

In terms of the previously introduced regularized mean problem, this striking result can be interpreted as a regularized solution in which the coefficient $\left(1 - \frac{(n-2)\sigma^2}{\|X\|^2} \right)$ plays the role of λ .

Another important consequence of the results in [85] is that, for $n = 1$, the ‘unregularized’ estimator $\bar{X} = X$ is “admissible” for every value of μ ; “admissible estimator” means “best possible estimator” with respect to (3.15).

The regularized mean problem is analogous to a 1-mean problem where the unique sample is \bar{x}_0 . In other terms:

1. Define one unknown parameter, the mean μ and correspondingly a Gaussian density function with variance σ^2/m (where m is the number of samples as per (3.1)).
2. Pick one sample from that p.d.f. and call it \bar{x}_0

James-Stein theory states that it does not exist an estimator for μ , linearly-based on \bar{x}_0 , which works better than \bar{x}_0 itself for $\mu \in [-\infty, +\infty]$ and taking $\mathbf{E}_X \{(\bar{x} - \mu)^2\}$ as quality measure.

However, thinking \bar{x}_0 as the sample mean derived from m samples of the same Gaussian p.d.f. with variance σ^2 , one can wonder if, for a finite range of μ , and using possibly a non linear elaboration of the samples, a better result than the non regularized one provided by \bar{x}_0 can be obtained.

In other words one can try to find an estimate \bar{x} such that $\mathbf{E}_X \{(\bar{x} - \mu)^2\} \leq \mathbf{E}_X \{(\bar{x}_0 - \mu)^2\}$ at least for μ in a predefined range (next section will clarify on this issue). Within this intuitive view one can try to apply classical statistical theory in the attempt of finding a value $\bar{x} = \lambda \bar{x}_0$ that mimics (3.12) as much as possible.

Summarizing the above considerations one has that:

1. The regularized mean problem can be linked to Stein theory.
2. Stein theory says that for $\mu \in [-\infty, +\infty]$ it is not possible to get a better estimation of μ than that provided by \bar{x}_0 , if only a linear \bar{x}_0 -based estimator is used.
3. Intuitively for a limited and pre-defined range of μ one can address the problem to obtain, from a set of m samples, an estimator for μ better than \bar{x}_0 . The problem is to elaborate samples in order to obtain estimates of μ useful for small values of m , i.e., when \bar{x}_0 can be unreliable. In particular the critical situations occur when m and μ^2/σ^2 are both small.

Given these observations the following part discusses some fundamental properties of the regularized mean problem with particular attention on the prior knowledge on μ .

3.1.2 Fundamental Laws of the Regularized Mean Problem

An important aspect concerns the study of effectiveness of regularization on the entire space when one has no prior knowledge on the distribution of μ .

Such a problem leads to analyze the asymptotic properties of:

$$\mathbf{E}_\mu \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\} \quad (3.17)$$

where the dependence of λ from the sample X has been underlined. In particular one has to inquiry on the relationship between (3.17) and the non regularized cost $\mathbf{E}_\mu \mathbf{E}_X \{(\bar{x}_0 - \mu)^2\} = \sigma^2/m$. A possible way to study (3.17) is to assume a uniform prior $p(\mu)$ over the range $[-\gamma, \gamma]$ and then taking the limit $\gamma \rightarrow \infty$ (for which the prior becomes improper): the convention \mathbf{E}_μ^γ will be used to indicate an expectation integral computed in the interval $[-\gamma, \gamma]$. Formally one has to study the following multiple integral:

$$\begin{aligned} & \mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\} = \\ & = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \int_{-\gamma}^{+\gamma} [\lambda(X)\bar{x}_0 - \mu]^2 \frac{1}{2\gamma} \frac{1}{(\sigma\sqrt{2\pi})^m} \prod_{i=1}^m \exp\left[-\frac{(x_i - \mu)^2}{2\sigma^2}\right] d\mu dx_1 \dots dx_m \end{aligned} \quad (3.18)$$

The following theorem clarifies a fundamental aspect on (3.18)

Theorem 3.1.1. *For any $\lambda(X) < 1$ the term $\mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\}$ fulfils the following properties:*

1. *if $\gamma \rightarrow \infty$ then $\mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\} - \sigma^2/m = \mathbf{E}_\mu \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\} - \sigma^2/m \rightarrow 0$ from positive values.*
2. *if $\gamma \rightarrow 0$ then $\mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\} - \sigma^2/m < 0$.*
3. *It exists at least a value $\hat{\gamma}$ such that:*

$$(a) \text{ For } \gamma < \hat{\gamma}, \mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\} - \sigma^2/m < 0$$

$$(b) \text{ In } \gamma = \hat{\gamma}, \mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\} - \sigma^2/m = 0$$

$$(c) \text{ For } \gamma > \hat{\gamma}, \mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\} - \sigma^2/m > 0$$

Proof. See Appendix

Point a) says that, whatever is the sample based regularization $\lambda(X)$ over all the range $[-\infty, +\infty]$, regularization is not effective. Thus it is not possible

to compute a regularizer in $[-\infty, +\infty]$ that always improves on the sample mean.

Point b), conversely, affirms that regularization when $\gamma \rightarrow 0$ is always convenient; this result is intuitive considering that $\lambda(X) < 1$ represents a shrinking of \bar{x}_0 towards 0.

The third result c) is the natural consequence of the two previous ones and states that for continuity (on γ) a finite range $[-\hat{\gamma}, +\hat{\gamma}]$ must exist where regularization is always effective.

The above theorem explicitly links the notion of a priori knowledge on $\hat{\gamma}$ parameter to the effectiveness of regularization. The more the interval $[-\hat{\gamma}, +\hat{\gamma}]$ is tight, the more effective is the regularization. In practice, the problem is to pick the $\lambda(X)$ (either linear or non-linear) covering efficiently the widest $[-\hat{\gamma}, +\hat{\gamma}]$ interval.

The behavior of (3.17) can be further evidenced considering the result it gives when the non sample based oracular value of λ is adopted.

Lemma 3.1.3. *Given $\mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda^{orac} \bar{x}_0 - \mu)^2\}$ and for any γ , the inequality $\mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda^{orac} \bar{x}_0 - \mu)^2\} \leq \sigma^2/m$ always holds true. In particular if $\gamma \rightarrow \infty$ then $\mathbf{E}_\mu \mathbf{E}_X \{(\lambda^{orac} \bar{x}_0 - \mu)^2\} - \sigma^2/m \rightarrow 0$ from negative values.*

This result confirms that the oracular regularizer (that is not sample based) is always useful when γ is finite.

In order to obtain a regularization effective for any γ , an ideal regularizer should depend both on μ and on the sample X : $\lambda^{ideal} = \frac{\mu}{\bar{x}_0}$. This fact, that may seem trivial, shows that effectiveness of regularization not only depends on oracular properties but also on the *sample properties themselves*: one more time the nature of regularization, as a sample-dependent correction strategy for a mathematical model, emerges.

As a further explanation of the involving phenomena one can show what happens when a not completely agnostic prior on μ is used. The following result shows that the a priori knowledge on the sign of μ grants the existence of an always effective non linear regularizer.

Lemma 3.1.4. *For $\lambda(X) = 1(\bar{x}_0)$, given $\mathbf{E}_\mu^\gamma \mathbf{E}_X \{(1(\bar{x}_0)\bar{x}_0 - \mu)^2\}$ and the prior $p(\mu) = 1(\mu) - 1(\mu - \gamma)$, the inequality $\mathbf{E}_\mu^\gamma \mathbf{E}_X \{(1(\bar{x}_0)\bar{x}_0 - \mu)^2\} \leq \sigma^2/m$ holds for*

any value of γ . In particular for $\gamma \rightarrow \infty$ then $\mathbf{E}_\mu \mathbf{E}_X \{(1(\bar{x}_0)\bar{x}_0 - \mu)^2\} - \sigma^2/m \rightarrow 0$ from negative values.

Proof. See Appendix.

Summing up, the above discussed results show that regularization with no prior knowledge on μ has some intrinsic limitations. These limits find full confirmation in Stein theory. The main outcome of the previous analysis is that searching for an efficient $\lambda(X)$ must be done without pretending to cover the whole range $\mu \in [-\infty, +\infty]$ but only a limited one. The following sections investigate on classical approaches such as leave one out [87] and evidence maximization, typical under Bayesian approach [88], to predict λ . Further the analysis is carried with a typical machine learning approach: feature selection followed by a learning paradigm.

3.1.2.1 The Leave One Out approach

This approach can be outlined by first considering the regularized mean value $\bar{\xi}_i$ obtained by removing from X the sample x_i . Taking into account (3.5), the $(m - 1)$ -elements regularized mean $\bar{\xi}_i$ is:

$$\bar{\xi}_i(\alpha) = \frac{\sum_{i \neq k, k=1}^m x_k}{m + \alpha - 1} \quad (3.19)$$

The sum of the m square error terms $(x_i - \bar{\xi}_i)^2$ yields the leave one out error functional:

$$L_{loo}(\alpha; X) \equiv \sum_{i=1}^m (x_i - \bar{\xi}_i)^2 \quad (3.20)$$

which can now be minimized with respect to α . Setting aside the mathematical details, which are reported in Appendix, the minimization procedure on $L_{loo}(\alpha; X)$ leads to obtain, for the regularization parameter α , the optimal value α^{loo} reported in Appendix and to write the corresponding value λ^{loo} .

Theorem 3.1.2. *The leave one out optimal regularizer λ^{loo} is:*

$$\lambda^{loo} = \frac{\bar{x}_0^2 - \frac{S^2}{m}}{\bar{x}_0^2} \quad (3.21)$$

where $S^2 = \frac{1}{m-1} \sum_{i=1}^m (x_i - \bar{x}_0)^2$. *Proof.* See Appendix

This estimator carries some possible inadequacies: the most evident is that λ^{loo} can become negative, thus changing the sign of the sample mean estimation. Another key point of this result is that λ^{loo} depends only on \bar{x}_0 , owing to the expression of S^2 . All information concerning X , except mean and variance estimation, is ignored. According to Stein theory such an estimator cannot grant a proper regularized solution because it based only on \bar{x}_0 .

3.1.2.2 Maximal Evidence

Evidence Maximization has been successfully used in Bayesian approach to neural networks [88]; here Evidence Maximization is applied to the regularized mean problem in order to find the regularization parameter.

As a first step, functional \mathfrak{S} in (3.1) is proposed into a generalized form usual in the Maximal Evidence approach:

$$\aleph(\xi; X, \tau, \varphi) \equiv \frac{\tau}{2} \sum_{i=1}^m (x_i - \xi)^2 + \frac{\varphi}{2} \xi^2 = \tau E_D + \varphi E_W \quad (3.22)$$

clearly, $\aleph(\xi; X, \tau, \varphi) \equiv \frac{\tau}{2} \mathfrak{S}(\xi; X, \frac{\varphi}{\tau})$. Now the optimal value of ξ for the *a posteriori* maximization of the functional (3.22) can be obtained.

Lemma 3.1.5. *The maximum a posteriori of the \aleph functional is obtained for:*

$$\bar{x}_{MP} = \arg \min_{\xi} \aleph(\xi; X, \tau, \varphi) = \frac{m\bar{x}_0\tau}{\varphi + m\tau} \quad (3.23)$$

Proof. See Appendix

Accordingly the maximum *a posteriori* cost value is $\aleph^{MP} = \varphi E_W^{MP} + \tau E_D^{MP}$. With these premises the following result holds:

Theorem 3.1.3. *Assuming a Gaussian prior, the optimal maximal evidence regularizer λ^{me} is:*

$$\lambda^{me} = \frac{\bar{x}_0^2 - \frac{S^2}{m}}{\bar{x}_0^2} \quad (3.24)$$

Proof. See Appendix

Quite surprisingly, then, $\lambda^{me} = \lambda^{loo}$. As a conclusion, maximization of evidence is equivalent to leave-one-out for this problem and shares the same limitations.

3.1.3 A Machine Learning Approach to the Mean Value Estimation

3.1.3.1 General Setting

The previous analysis shed some light on the fact that, according to Stein theory [85], classical methods give the regularization parameter a value which is unsatisfactory in terms of the quality metric (3.10). Now the aim is to analyze the problem with the usual machine learning perspective; in other terms, a feature extraction step followed by a training process of a learning machine. In this study Multiple Kernel Learning [51] and Singular Value Decomposition are the tools adopted for features extraction, while the training process uses alternatively a neural network and a rational function to obtain an estimate \bar{x} of μ more accurate than \bar{x}_0 . According to the previous considerations, the feature selection and the training process has been carried considering the most important case of sample means \bar{x}_0 generated by a small number m of samples and within a limited range of μ^2/σ^2 .

3.1.3.2 Data Generation for Multiple Kernel Learning

In order to obtain a near *oracular* regularizer, the λ^{orac} term has been defined as target and the problem thought as a regression problem.

For any m -vector $X \sim N(\mu, \sigma^2)$ of samples, the corresponding input data pattern has been defined as a vector v made up of $2m$ real numbers: the first m elements are the sample values x_i organized in ascending order, the remaining elements are the squares of the previous ones. Then the structure of the input vector v is:

$$\mathbf{v} = \{x_{MIN}, \dots, x_{MAX}, x_{MIN}^2, \dots, x_{MAX}^2\} \quad (3.25)$$

This square preprocessing is reminiscent of the features used as input for the circular back propagation network [66]. The target is λ^{orac} and can be computed by (3.12).

One generates a data matrix \mathbf{V} according to the following strategy:

1. Fix the values of σ and m .

2. Set a range for μ and a uniform sampling of this range where each element is μ_i and the number of elements is n_μ ; correspondingly, compute each λ_i^{orac} . For each μ_i generate a set of m random elements distributed as $N(\mu, \sigma^2)$. Then write the corresponding v_i as in (3.25).

The quantity v_i becomes the i -th row of the data matrix V of size $n_\mu \times 2m$, which is associated to a vector of targets λ whose n_μ elements are λ_i^{orac} .

3.1.3.3 Multiple Kernel Learning

Multiple Kernel Learning [51] is a powerful technique where, differently from SVM, the kernel matrix is a convex superposition of single kernel matrices. In particular the primal problem can be formulated through a generalization of the usual functional involved in SVM for regression. The functional formulation needs to introduce this set of terms:

1. n_k is the number of kernels (in this case $n_k = 2m$ as will be later explained)
2. d is the vector of size n_k whose j -th element d_j is the weight of the j -th kernel
3. \hat{f}_j is the learned function corresponding to the j -th kernel only. Denoting by v_i the usual expansion coefficients (based on Representer Theorem [48]) one has:

$$\hat{f}_j(v) = \sum_{i=1}^{n_\mu} v_i K_j(v, v_i) \quad (3.26)$$

4. The function $f(v)$, that should estimate λ^{orac} , is taken as a weighted sum of the \hat{f}_j 's plus a bias term b according to following expression:

$$f(v) = \sum_{j=1}^{n_k} d_j \hat{f}_j(v) + b \quad (3.27)$$

Defining $f_j(v) = d_j \hat{f}_j(v)$, the minimization problem to be solved is:

$$\left\{ \begin{array}{l} \min_{d,v,b,\xi_i} \frac{1}{2} \sum_{j=1}^{n_k} \frac{1}{d_j} \|f_j\|^2 + C \sum_{i=1}^{n_\mu} (\xi_i + \xi_i^*) \\ \text{s.t.} \quad \lambda_i^{orac} - \sum_{j=1}^{n_k} f_j(v_i^{(x)}) - b \leq \varepsilon + \xi_i \quad \forall i \\ \quad \quad -\lambda_i^{orac} + \sum_{j=1}^{n_k} f_j(v_i^{(x)}) + b \leq \varepsilon + \xi_i^* \quad \forall i \\ \quad \quad \xi_i \geq 0, \xi_i^* \geq 0 \quad \forall i, \sum_{j=1}^{n_k} d_j = 1, d_j \geq 0 \quad \forall j \end{array} \right. \quad (3.28)$$

Where ξ_i, ξ_i^* are slack variables, ε is the insensitivity parameter [3] and C is the (inverse) regularization parameter.

In the dual formulation of the problem (3.28) the solution is obtained via a reduced gradient method getting the coefficients d_j of the kernel matrices together with the expansion coefficients v_i of the learned function [51].

In order to get a feature selection, one takes a kernel for each feature. This amounts to work with exactly $n_k = 2m$ kernels all with identical parameters (e.g. width for Gaussian kernel). Thanks to this choice, the magnitude of the coefficients d_j represents the importance of the j -th kernel (feature) in the prediction of λ^{orac} . In other words, a high value of d_j means that j -th feature is important in the prediction of λ^{orac} , while small values of d_j mean that j -th feature is not relevant.

For a statistically meaningful result one has to assess what are, *on the average*, the features that most influence the prediction function. To this end one can operatively re-run MKL several times and then consider the vector \bar{d} , defined as the average of the different vectors d . The d vectors derive from different runs on different randomly generated data as in the data generation pseudo-code.

In procedural terms, then, one must consider the following steps:

1. Set a vector \bar{d} of size $2m$ all to 0.
2. Generate a data matrix V and targets λ according to the described data generation procedure
3. Perform MKL regression on V and λ . This procedure outputs a vector d .
4. Compute $\bar{d} \leftarrow \bar{d} + d$

5. Repeat 2)-4) P times
6. The average weights vector is computed as $\bar{d} \leftarrow \bar{d}/P$.

This procedure avoids using a single run of MKL with a high number of patterns, operation that can be computationally demanding [51].

The kernels chosen for this analysis are Gaussian kernels. All of them share the same width parameter which can simply be set to the variance σ of the data. In this context Over-fitting is not a major problem because one performs training using the entire population; in this case generalization performance of the machine is not a concern.

RBF width and data variance σ are set to 1; C parameter value is 100; size m of the samples X is 10 and $P = 1e3$. The software used is the publicly available SimpleMKL [51]. Figure 3.2 represents the results obtained generating data in the range $\mu \in [0, 1]$ sampled with a step of 0.02 (50 patterns for each run). The values of the \bar{d} components are plotted vs. their positional indexes, which correspond to the indexes of the components of v . The results evidence some relevant features:

1. Curve on figure 3.2 exhibits two branches, the first of which is concave, and the other convex. The maximum of the concave branch corresponds to the sample closest to the sample mean \bar{x}_0 .
2. In the convex branch, maxima are attained at the extremes x_{MIN}^2 and x_{MAX}^2 .
3. The \bar{d}_i weight at x_{MAX}^2 is greater than at x_{MIN}^2 . This is due to the asymmetry of the range $\mu \in [0, 1]$. Fig. 3.3 deals with the results obtained in the range $\mu \in [-1, 0]$ and shows the expected swap of the roles of x_{MAX}^2 and x_{MIN}^2

This first step of analysis gives an important hint on the significant sample values that should be used in a regularizer predictor: these values are x_{MAX}^2, x_{MIN}^2 , or equivalently $|x_{MAX}|^2, |x_{MIN}|^2$, and \bar{x}_0 .

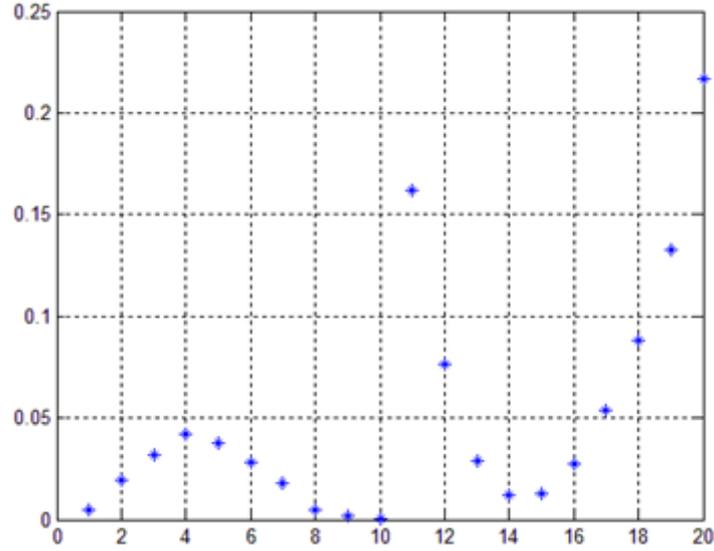


Figure 3.2: Multiple Kernel Learning average kernel weights values for $\mu \in [0, +1]$: x axis is each feature (kernel) and y axis is the average associated weight

3.1.3.4 The oracular gap: an analysis based on Singular Value Decomposition

In this section the oracular gap will be introduced and studied. This quantity represents the difference between the oracular regularized mean $\lambda^{orac}\bar{x}_0$ and the sample mean \bar{x}_0 . The oracular gap value, when thought as the target for a function g of the elements in \mathbf{X} , gives another perspective by which the feature selection problem can be studied. This inquiry aims at a deeper insight into the results of the previous analysis through a completely different mathematical tool that takes as target the *gap* instead of the oracular regularizer.

The oracular gap is defined by:

$$\delta^{orac} \equiv \lambda^{orac}\bar{x}_0 - \bar{x}_0 = -\bar{x}_0 \frac{\sigma^2}{m\mu^2 + \sigma^2} \quad (3.29)$$

The problem now is to find a function g predicting δ^{orac} . The function g is thought as linear with respect to its arguments, which derive from the k -th realization of m samples $x_1^{(k)}, \dots, x_m^{(k)}$. The arguments are $|x^{(k)}|_{MIN}^2 =$

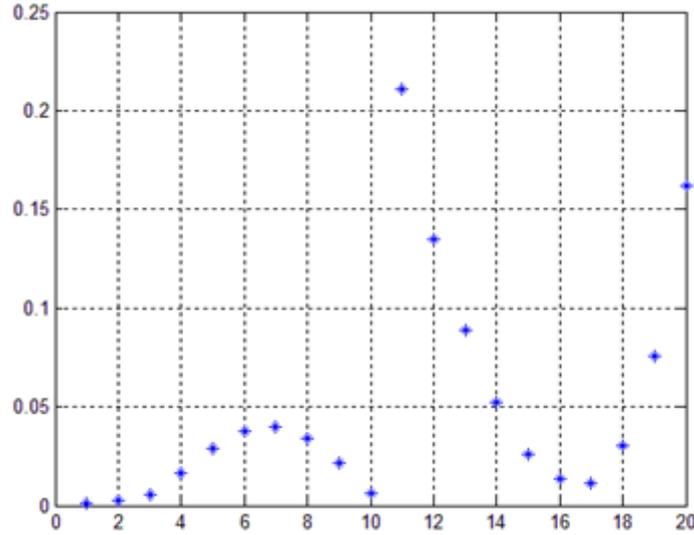


Figure 3.3: Multiple Kernel Learning average kernel weights values for $\mu \in [-1, 0]$ x axis is each feature (kernel) and y axis is the average associated weight

$\left(\min\{|x_1^{(k)}|, \dots, |x_m^{(k)}|\}\right)^2$, $|x^{(k)}|_{MAX}^2 = \left(\max\{|x_1^{(k)}|, \dots, |x_m^{(k)}|\}\right)^2$. Then the structure assumed for g is:

$$g(|x|_{MAX}^2, |x|_{MIN}^2) = b_1 + b_2|x|_{MAX}^2 + b_3|x|_{MIN}^2 \quad (3.30)$$

Taking b_1, b_2, b_3 as the components of a coefficient vector b and denoting by \mathbf{X} the matrix whose k -th row is $1, |x^{(k)}|_{MAX}^2, |x^{(k)}|_{MIN}^2$ the problem to be solved corresponds to the linear system:

$$\mathbf{Xb} = \mathbf{y} \quad (3.31)$$

where each component of vector \mathbf{y} is $y_k = -\bar{x}_0^{(k)} \frac{\sigma^2}{m\mu^2 + \sigma^2}$. Since system (3.31) is defined by a 'slim' matrix \mathbf{X} , \mathbf{b} is the least square solution vector.

The three components of the least square solution \mathbf{b} contribute to estimate the target vector \mathbf{y} in a well defined way, which can be evidenced by considering the following steps based on the Singular Value Decomposition of the matrix \mathbf{X} :

1. The SVD of \mathbf{X} can be written as $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^t = \sum_{i=1}^3 s_i u_i (\mathbf{v}_i)^t$
2. It follows that $\mathbf{X}\mathbf{b} = \sum_{i=1}^3 s_i \langle \mathbf{v}_i, \mathbf{b} \rangle \mathbf{u}_i$
3. The dominant element in the sum is $s_1 \langle \mathbf{v}_1, \mathbf{b} \rangle \mathbf{u}_1$ (where s_1 is the largest singular value). Explicitly one has $s_1 \langle \mathbf{v}_1, \mathbf{b} \rangle = s_1 [v_1(1)b_1 + v_1(3)b_2 + v_1(3)b_3]$

This scheme allows to look for the importance of the terms $|x^{(k)}|_{MAX}^2$ and $|x^{(k)}|_{MIN}^2$ (associated respectively to b_2, b_3) in the prediction of \mathbf{y} : in particular the terms $|s_1 b_2 v_1(2)|$ and $|s_1 b_3 v_1(3)|$ are considered. Some general information concerning these contributions can be obtained taking the average values w_{MAX} and w_{MIN} of the terms $|s_1 b_2 v_1(2)|$ and $|s_1 b_3 v_1(3)|$ generated from different matrices \mathbf{X} .

For given values of σ, μ and m , the average behavior of the terms $|s_1 b_2 v_1|, |s_1 b_3 v_1|$ is obtained by the following procedure:

1. Set scalars $w_{MAX} = 0, w_{MIN} = 0$
2. Generate \mathbf{X} for the given set of parameters.
3. Solve linear regression problem (3.31)
4. Compute SVD of \mathbf{X}
5. $w_{MAX} \leftarrow w_{MAX} + |s_1 b_2 v_1(2)|$ and $w_{MIN} \leftarrow w_{MIN} + |s_1 b_3 v_1(3)|$
6. Repeat from 2) to 5) P times
7. Output: $w_{MAX} \leftarrow w_{MAX}/P$ and $w_{MIN} \leftarrow w_{MIN}/P$

Resulting w_{MAX} and w_{MIN} give a estimate of the importance of $|x^{(k)}|_{MAX}^2$ and $|x^{(k)}|_{MIN}^2$.

In figure 3.4 we used $\sigma = 1, m = 10$, the number of rows of \mathbf{X} at each run was $10^4, P=10$ and the range $\mu \in [-1, 1]$ (step 0.1) was analyzed: this range is the same range used in the previous analysis performed with MKL.

The obtained results, plotted in fig. 3.4, evidence that:

1. The influence term w_{MAX} is always higher than w_{MIN} , as expected from the previous analysis and this result stresses the importance of $|x|_{MAX}^2$.

2. Both influence terms w_{MAX} and w_{MIN} show an abrupt decrease when $\mu \cong 0$.
3. For sufficiently large values of $|\mu|$, w_{MAX} , w_{MIN} and the gap ($w_{MAX} - w_{MIN}$) decrease as $|\mu|$ increases. This means that the role of $|x_{MAX}^2|$ and $|x_{MIN}^2|$ in the oracular gap estimation becomes less important as $|\mu|$ increases, as expected.

Other settings for this analysis lead to analogous results.

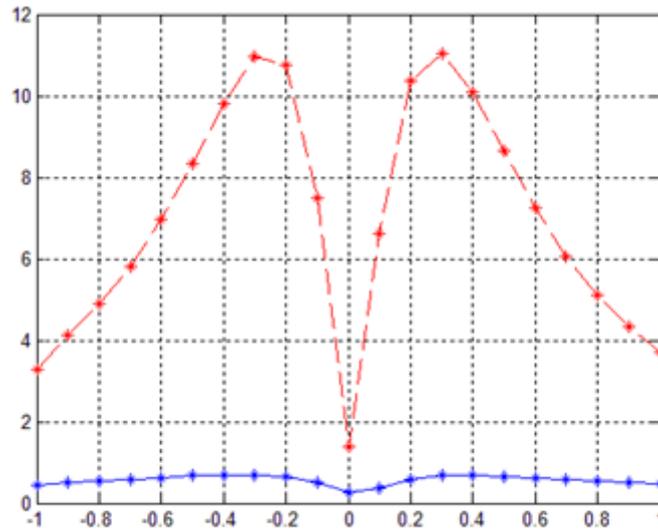


Figure 3.4: Dashed line is w_{MAX} and w_{MIN} is continuous line: x axis is μ value and y axis are w_{MAX} and w_{MIN} .

3.1.3.5 Some summarizing comments

The above results evidence the important role played by the terms $|x_{MAX}|^2$, $|x_{MIN}|^2$ and \bar{x}_0 in the prediction of the oracular regularizer. This role was evidenced through the MKL analysis and it is confirmed by the SVD-based discussion, which also yields additional information about the behavior of $|x_{MAX}^2|$ vs. μ and its dominant role with respect to $|x_{MIN}|^2$ in predicting λ^{orac} . As a conclusion, the terms \bar{x}_0 and $|x_{MAX}^2| = \max(|x_{MAX}|, |x_{MIN}|)^2$ are the most reasonable input elements of a nonlinear mean value estimator.

3.1.3.6 Rational non linear regularizers for mean value estimation

In order predict a regularizer one has to cope with the quality metric:

$$y = \mathbf{E}_X \{(\bar{x}_0 - \mu)^2\} - \mathbf{E}_X \{(\lambda \bar{x}_0 - \mu)^2\} \quad (3.32)$$

Observe that a positive value of y means that regularization gives a better estimation, while $y \leq 0$ means that regularization is useless ($y = 0$) or deteriorates the estimation ($y < 0$).

Owing to Stein theory, a better estimation of \bar{x} cannot be simply based on \bar{x}_0 . Following [89] one can approximate λ^{orac} with $\hat{\lambda}^{orac}$ obtained by replacing μ with \bar{x}_0 in expression (3.12). The continuous line in figure 5 represents expression (3.32) vs $\mu \in [-1, +1]$ for $m=1, \sigma = 1$ using $\hat{\lambda}^{orac}$.

From leave-one-out, or equivalently from maximal evidence, analysis one can define:

$$\lambda_{ht}^{loo} = \max \left(0, \frac{\bar{x}_0^2 - \frac{\sigma^2}{m}}{\bar{x}_0^2} \right) \quad (3.33)$$

The modification of (3.21) is due to the non negativity constraint on λ .

Also this regularizer leads to inadequate solutions (see dashed line on 3.5). It can be observed that these regularizers ($\hat{\lambda}^{orac}, \lambda_{ht}^{loo}$) are very similar and both have wide areas in which the non regularized solution is better than the regularized counterpart.

Other non linear regularizers could be considered: in particular one can use $|x|_{MAX}^2$, as suggested in the conclusions of the previous section. Indicating for simplicity of notation $\beta^2 \equiv |x|_{MAX}^2$ one can define two new regularizers.

The first is obtained by substituting \bar{x}_0^2 with β^2 in (3.21):

$$\lambda_{\beta}^{loo} = \frac{\beta^2 - \frac{\sigma^2}{m}}{\beta^2} \quad (3.34)$$

while the second is obtained by substituting \bar{x}_0^2 with β^2 in $\hat{\lambda}^{orac}$

$$\lambda_{\beta}^{orac} = \frac{\beta^2}{\beta^2 + \frac{\sigma^2}{m}} \quad (3.35)$$

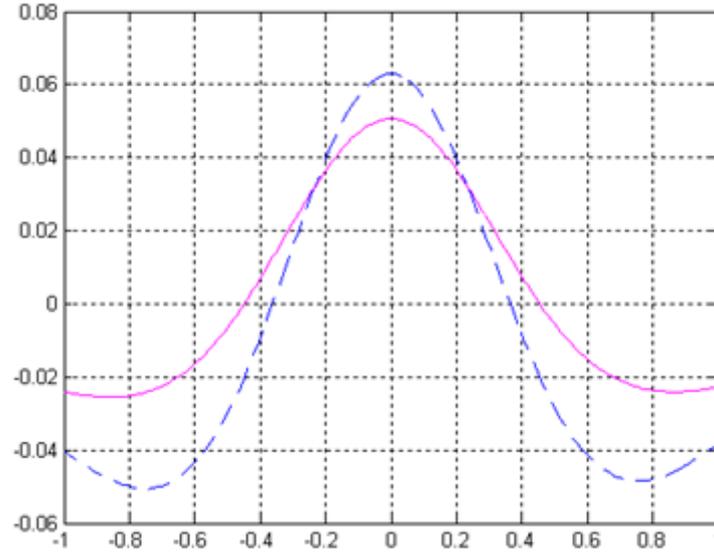


Figure 3.5: x axis is μ value and y axis is the quality metric (3.32). Continuous line stands for $\hat{\lambda}^{orac}$, and dashed line for λ_{ht}^{loo}

The rationale behind this choice is twofold: first, β^2 has been selected as a candidate variable by the previous analysis, secondly, this choice produces a conservative regularizer. Regularizers λ_{β}^{loo} and λ_{β}^{orac} are much closer to one than λ^{loo} and $\hat{\lambda}^{orac}$ respectively. Then the regularized solutions $\lambda_{\beta}^{loo}\bar{x}_0$ and $\lambda_{\beta}^{orac}\bar{x}_0$ are closer to \bar{x}_0 than $\lambda^{loo}\bar{x}_0$ and $\hat{\lambda}^{orac}\bar{x}_0$ respectively. So, both regularizers λ_{β}^{loo} and λ_{β}^{orac} are conservative because one is using $|x|_{MAX}^2$ instead of \bar{x}_0 . A conservative approach leads to a less efficient regularization in terms of expression (3.32) but makes possible to extend the predictor efficiency to greater values of μ^2/σ^2 .

Finally, by replacing σ^2 with S^2 in (3.35) another estimator based only on the sample is considered:

$$\hat{\lambda}_{\beta}^{orac} = \frac{\beta^2}{\beta^2 + \frac{S^2}{m}} \quad (3.36)$$

In figure 3.6 ($m=10, \sigma^2 = 1, \mu \in [-1, +1]$) λ_{β}^{loo} , λ_{β}^{orac} and $\hat{\lambda}_{\beta}^{orac}$ are studied in terms of the quality measure (3.32). The behavior is very similar and both $\hat{\lambda}_{\beta}^{orac}$ and λ_{β}^{loo} can be reasonably used as practical regularizer because they are

completely sample-based and no knowledge on σ^2 is required.

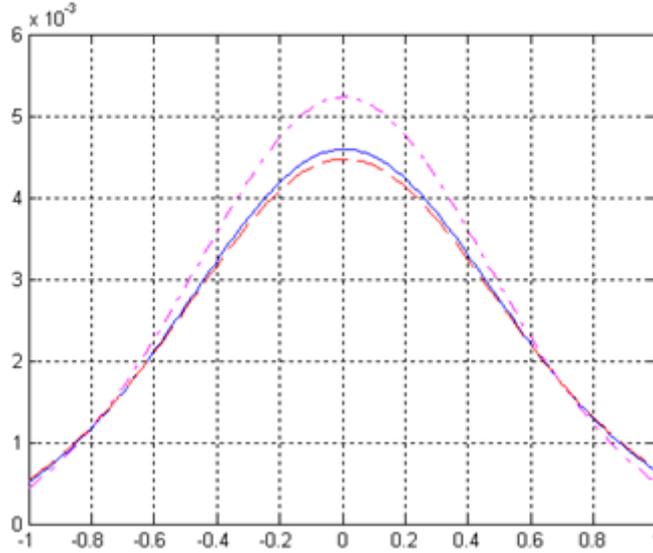


Figure 3.6: Point-Dashed line for λ_{β}^{orac} , continuous line is for λ_{β}^{loo} and dashed line is for $\hat{\lambda}_{\beta}^{orac}$: x axis is μ value and y axis is the quality metric (3.32)

Comparing figure 3.5 and 3.6 one can appreciate the advantage of using β^2 in $\lambda_{\beta}^{orac}, \lambda_{\beta}^{loo}$ and $\hat{\lambda}_{\beta}^{orac}$ with respect λ_{ht}^{loo} and $\hat{\lambda}^{orac}$ that both use \bar{x}_0^2 . Figure 3.6 shows the entire family of functions (3.32) where λ_{β}^{loo} is used with varying σ^2 ; if one *a priori* knows that, for a given m , the term μ^2/σ^2 lies inside the positive region of curve in 3.7, then regularization can be used; otherwise regularization is not the proper choice.

Two interesting aspects emerge from the analysis of $\lambda_{\beta}^{loo}, \hat{\lambda}_{\beta}^{orac}$:

1. These regularizers exhibit a linear dependence of the range of μ individuated by the condition $y = \mathbf{E}_X \{(\bar{x}_0 - \mu)^2\} - \mathbf{E}_X \{(\lambda \bar{x}_0 - \mu)^2\} > 0$, with respect to σ . Figure 3.7 shows the behavior of y vs μ when using λ_{β}^{loo} for values of the σ parameter in the range $[0.2, 2]$ (step 0.2), $m=10$ and $\mu \in [-5, +5]$. Almost identical curves could be obtained by replacing λ_{β}^{loo} with $\hat{\lambda}_{\beta}^{orac}$.
2. Another intriguing feature concerns the behavior of the peaks in Figure 3.7, because the peak values pk versus σ can be precisely represented by

the parabola $pk = 0.0046\sigma^2$ (analogous results hold for different parameters).

The above results indicate regularity on the behavior of the quality metric (3.32) with respect to σ when employing the proposed regularizers.

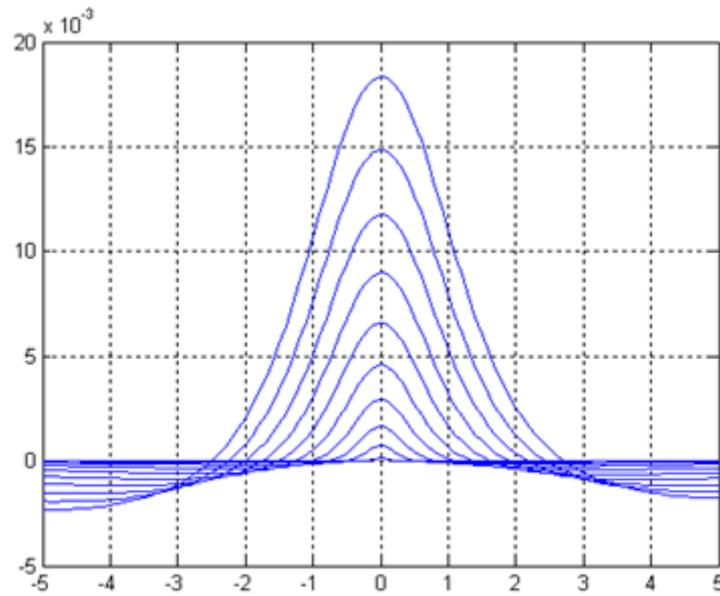


Figure 3.7: Behavior of the quality metric y when using λ_{β}^{loo} : x axis is μ value with varying sigma in the range $[0.2, 2]$ (step 0.2). Highest peak curve corresponds to the highest value of the parameter σ

3.1.3.7 Expected Value Estimation by a Neural Network

In order to define a neural model for the expected value μ , some terms and conditions have to be defined:

1. Suppose, as hypothesis, to know the range $\mu \in [-\mu_M, +\mu_M]$ and the value of σ .
2. Suppose using a single hidden layer neural network.
3. Generate several realizations distributed as $N(\mu, \sigma^2)$. Set as input of the NN the values \bar{x}_0 and β^2 derived from these realizations.

4. Call g the output function of the NN; then the prediction model is:

$$g\mu_M + \bar{x}_0 = \bar{x} \quad (3.37)$$

The NN has to predict the offset by which \bar{x}_0 should be modified to improve the estimation of μ

1. Set as target values the normalized offset $\frac{\mu - \bar{x}_0}{\mu_M}$
2. Set as cost function

$$L(\bar{x}_0, \mu_M, \mu) = \left(\frac{\mu - \bar{x}_0}{\mu_M} - g \right)^2 \quad (3.38)$$

The final goal is trying to minimize $\mathbf{E}_X (\mu - \bar{x})^2$ by using (3.38) in a predefined range of μ .

This aspect can be clarified considering that:

$$L(\bar{x}_0, \mu_M, \mu) = \left(\frac{\mu - \bar{x}_0}{\mu_M} - g \right)^2 = \frac{1}{\mu_M^2} \mathbf{E}_X (\mu - (g\mu_M + \bar{x}_0))^2 = \frac{1}{\mu_M^2} \mathbf{E}_X (\mu - \bar{x})^2 \quad (3.39)$$

This minimization process can be accomplished by a classical back propagation algorithm. Moreover, due to the number of involved patterns, an on-line version of back-propagation has been used. The proposed network configuration is efficient over a wide range of the expected value μ . However, the expected symmetry of y around $\mu = 0$ in eq.(3.32) can suffer from some uncertainties, mainly due to the randomness of the learning process. In order to avoid this a balancing structure for the prediction has been employed (see fig.3.8). In this structure, two identical NNs are requested to predict on input pairs (\bar{x}_0, β^2) and $(-\bar{x}_0, \beta^2)$ producing the respective predictions g_1 and g_2 ; the final \bar{x} is computed as $((g_1 - g_2)/2)\mu_M + \bar{x}_0$. This expedient yields symmetric expectations over a wide range of μ .

Parameters for the experiments were:

1. A range for $\mu \in [-15, +15]$ by steps of 0.1, $\sigma^2 = 1$, and varying number of samples $m = [2, 14]$.

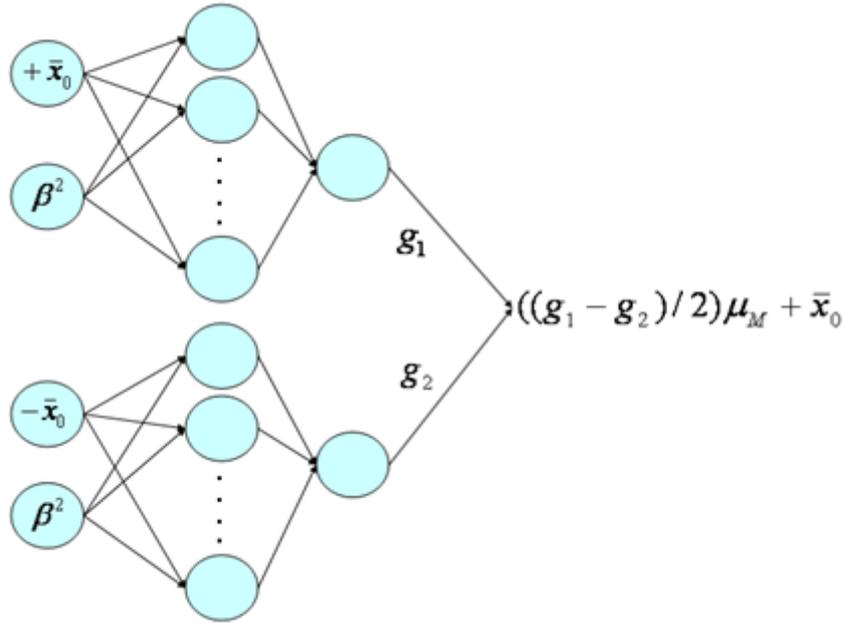


Figure 3.8: Neural scheme

2. The number of patterns was 10^6 and 10^5 for each μ value, for training and test respectively.
3. The learning rate was set to 10^{-4} .
4. The number of hidden neurons was 30.

A Matlab version of the software, available under request, has been implemented. It loads pre-computed weights and predicts the regularized mean value.

The following figures describe the outcomes using the quality measure (3.32); in particular in the range $\mu \in [-11, +11]$ regularization induced by the neural network is effective. This result shows that, working within an *a priori* known range and variance, regularization can be always useful; this outcome ultimately confirms the theory developed in previous sections on the prior over μ .

Given the prediction model $g\mu_M + \bar{x}_0 = \bar{x}$ it is possible to work out the equivalent neural regularizer $\lambda^{nn} = 1 + \frac{g\mu_M}{\bar{x}_0}$; experiments show that λ^{nn} can be

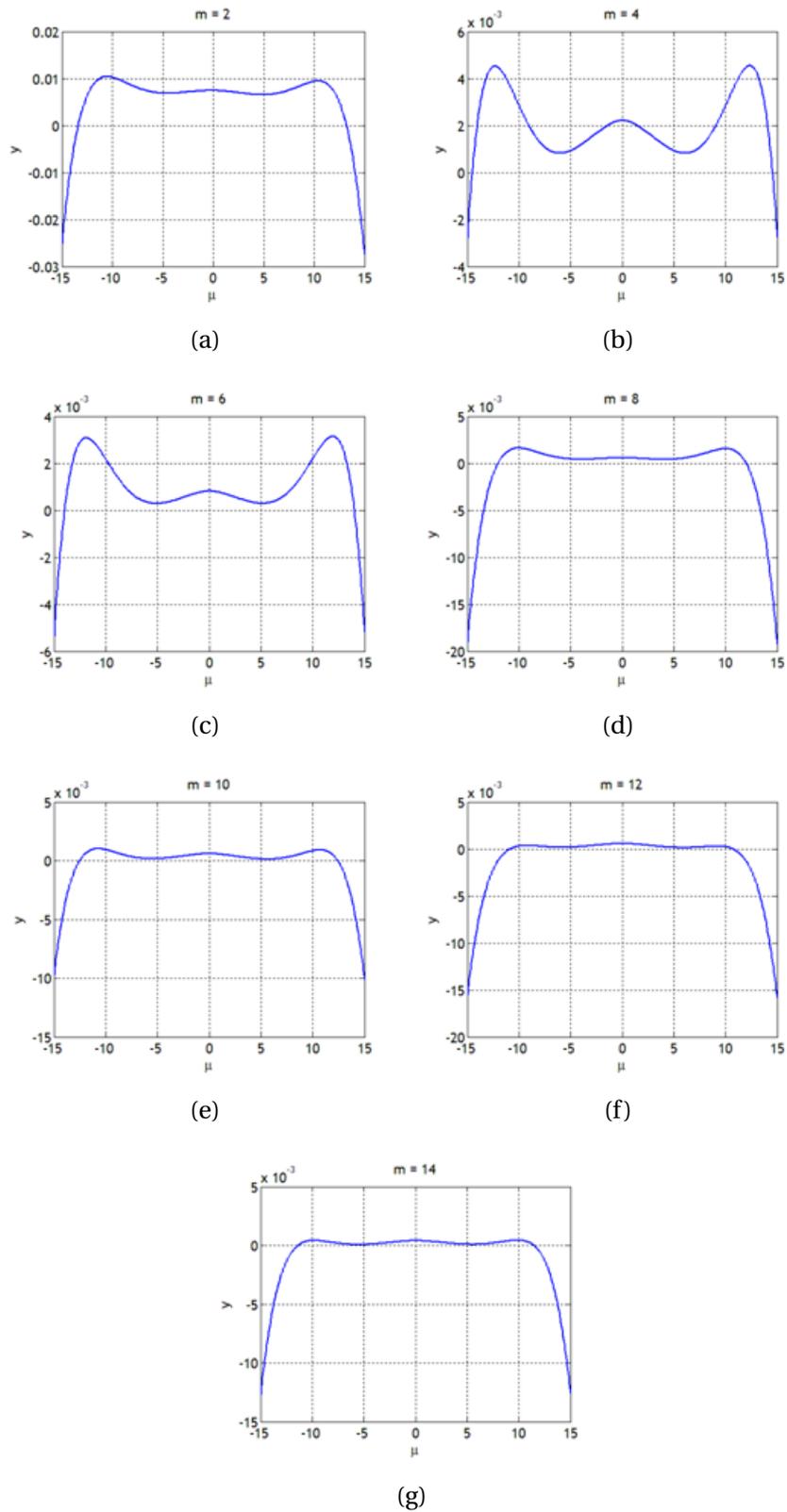


Figure 3.9: Neural Regularizers results: x axis is μ value and y axis is the quality metric

greater than 1. This fact is physiological of the neural network because there is no explicit mathematical constraint on λ^{mn} during the training process. This first group of experiments aimed at showing that a wide range of μ can be covered by the developed network. The second group of experiments deals with the effectiveness and usefulness of the regularized solution induced by the neural network when used in a relatively narrow range. One can suppose that μ_M represents a limit for a, possibly normalized, physical quantity (e.g. voltage, current etc...); thus one can define a normalized limit as $\mu_M = +1$ before training the neural network. The settings of the neural network are identical to those defined for the first group of experiments except for the number of neurons: after a preliminary model selection this number was set to 5.

As shown in 3.10, the percent gain computed as $\left(1 - \frac{\mathbf{E}_X\{(\lambda^n n \hat{x}_0 - \mu)^2\}}{\mathbf{E}_X\{(\hat{x}_0 - \mu)^2\}}\right) * 100$ is notable; its values strictly depend on the number of samples m ; the lower is m the higher is the gain. This outcome further stresses that regularization is extremely useful when one has a limited sample and a *a priori* knowledge of the problem is given, i.e. the range of μ . As a last experiment the case $m = 50$ was tested: figure 3.11 shows that the gain shrinks in a range defined by 2 the gain is still significant considering the high number of samples.

3.1.4 Conclusions

In this first study the regularized mean problem has been introduced and discussed. Theory showed that, according to James-Stein theory, mean prediction exhibits some intrinsic limitations. These limitations can be mitigated by a non linear approach to the problem and with a priori knowledge on the range of the mean value and/or the variance.

A numerical analysis has shown the role played by the biggest and the smallest sample on a sampling process. SVD and MKL together with a statistical analysis have been proposed to better understand the importance of each sample when estimating the expected value. A rational function and a neural network model have been proposed as viable tools to predict the mean; in particular the neural network model proved particularly effective. The next step is analyzing Tikhonov regularization.

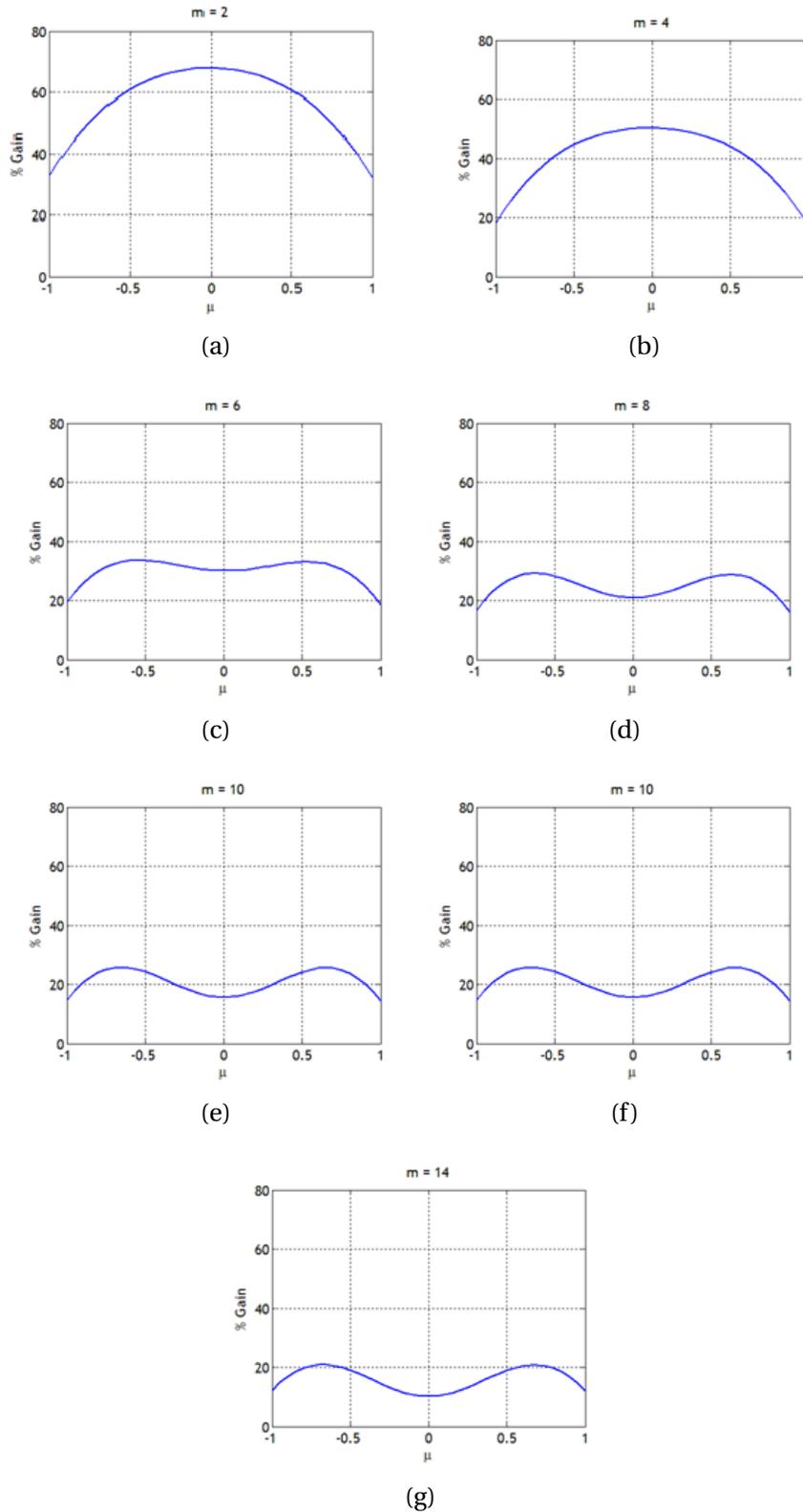


Figure 3.10: Neural Regularizers results on the domain $\mu_M = 1$: x axis is μ value and y axis is the quality metric

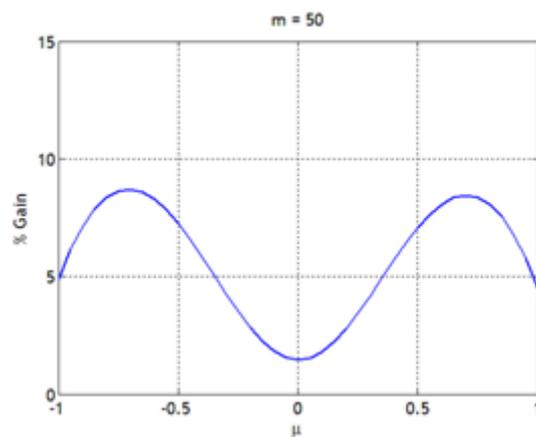


Figure 3.11: Neural Regularizers results: x axis is μ value and y axis is the percentual gain for $m = 50$

3.2 Generalized Tikhonov

In this work a unifying view of Learning, Regularization, Shrinking and Filtering is proposed. It is historically clear that such problems are often studied in diverse mathematical environments and superficially can appear as separate problems; indeed it will be shown that shrinking, filtering and learning are specific instances of regularization. A unique general theory presented here allows to give several insights.

A generalized Tikhonov regularization problem is defined, and the oracular closed form solution of the regularization matrix \hat{T} is derived. Throughout the paper the word *oracular* will mean that in expectation no better solution can be achieved.

The central result shows that using only one single regularization constant α is not sufficient to grant an oracular closed form solution. Among some intermediate results a suitable definition of degree of freedom is proposed.

The obtained results link the Regularization problem to Learning, Shrinking and Filtering. Form the Learning perspective it is shown that the developed machinery allows to define an ideal Universum of points for regression, where by Universum one indicates the concept of Universum introduced by Vapnik in [7]. Then a notion of oracular linear kernel is derived.

In the statistical environment it is shown that shrinking is a particular instance of Tikhonov regularization and that links between Copas Theory [90] and oracular regularization exist. Further the d-means problem is addressed and it is shown that Stein paradox, when dealing with oracular quantities, disappears; from the applicative view-point knowing the structure of oracular regularization allows to define new shrinkers (i.e. regularizers) different from the one proposed in James-Stein theory; quite surprisingly one of them in expectation is significantly better than James-Stein shrinker in a wide signal to noise ratio range when the sign of the mean vector is constant

Further it is proved that generalized Tikhonov regularization for circulant matrices (i.e. circular convolution operators) and oracular regularization is identical to Wiener filtering with ideal Signal to Noise Ratio estimation for each frequency of the spectrum.

A final discussion proposes a coherent way to define the encountered different levels of regularization and give some insights on a generalized class of shrinkers and filters.

3.2.1 Generalized Tikhonov and Oracular Regularization

Let $x_i \in \mathbb{R}^n$ m experimental observations each equipped with a label $y_i \in \mathbb{R}$. Let X the matrix $m \times n$ of the data, and y the vector of labels. The variable y is subject to a noise ε with zero mean and variance σ_y^2 . The matrix X is not subject to noise. The dependence relation between x and y is:

$$y = y_* + \varepsilon \quad (3.40)$$

$$y_* = Xw_* \quad (3.41)$$

where $w_* \in \mathbb{R}^n$ it is the optimal vector of coefficients. If the noise σ_y^2 is zero, w_* represents the obtained solution via pseudo-inversion $w_* = X^+y$. In the general case $\sigma_y^2 \neq 0$ then it is convenient to regularize the solution as per:

$$\hat{w} = \arg \min_w \{ \|y - Xw\|^2 + \alpha \|w\|^2 \} \quad (3.42)$$

where $\alpha > 0$ is the regularization constant. This formulation is known as ridge regression or Tikhonov regularization. In order to show some statistical properties of Tikhonov regularization it is necessary to slightly generalize the functional to :

$$\hat{w} = \arg \min_w \|y - Xw\|^2 + \|\hat{T}w\|^2 \quad (3.43)$$

where \hat{T} is a positive definite square matrix and is the generalized regularization operator where regularization parameters are embedded. In particular \hat{T} has the following structure:

$$\hat{T} = TR \quad (3.44)$$

Where R is a rotation matrix and T is a diagonal matrix composed by the

regularization entries $\theta_1, \theta_2, \dots, \theta_n$.

$$\begin{pmatrix} \theta_1 & 0 & 0 & \cdots & 0 \\ 0 & \theta_2 & 0 & \cdots & 0 \\ 0 & 0 & \theta_3 & & \vdots \\ \vdots & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \theta_n \end{pmatrix} \quad (3.45)$$

Using SVD on X one has that:

$$X = U\Sigma V^t \quad (3.46)$$

- U is an orthogonal matrix $m \times m$
- Σ is a diagonal matrix $m \times n$
- V is an orthogonal matrix $n \times n$

the diagonal values on Σ are the singular values and are indicated as $\sigma_1, \sigma_2, \dots, \sigma_r$ where $r = \text{rank}(X) \leq \{m, n\}$. Then one has:

$$X = \sum_{i=1}^r \sigma_i u_i v_i^t \quad (3.47)$$

Then the solution of the system 3.43 is:

$$\hat{w} = V(\Sigma^t \Sigma + V^t \hat{T}^2 V)^{-1} \Sigma^t U^t y \quad (3.48)$$

This formulation does not allow an easy manipulation; for this reason the SVD properties are used to simplify computations. First consider that w lives on the space spanned by the orthogonal vectors of the matrix V ; in other words one can define a base β where $w = V\beta$ is true and consequently every component $\beta_i = \langle w, v_i \rangle$, where \langle, \rangle denotes the scalar product. Then by definition of \hat{T} one sets the following relation:

$$\hat{T} = T V^t \quad (3.49)$$

With these premises the new solution vector is given by:

$$\hat{\beta} = \arg \min_{\beta} \{ \|y - U\Sigma\beta\|^2 + \|T\beta\|^2 \} \quad (3.50)$$

That is:

$$\hat{\beta} = A^{-1}\Sigma^t U^t y \quad (3.51)$$

where $A^{-1} = (\Sigma^t \Sigma + T^2)^{-1}$ is a $n \times n$ diagonal matrix where the main diagonal contains the values $\frac{1}{\sigma_i^2 + \theta_i^2}$; then indicate by $D_n(\cdot)$ a diagonal matrix of size n and entries (\cdot) ; thus $\left[D_n \left(\frac{1}{\sigma_i^2 + \theta_i^2} \right) \mid 0_{n,m-n} \right] = A^{-1}\Sigma^t$. From this property one can obtain the solution $\hat{w} = V\hat{\beta}$.

$$\hat{w} = V \left[D_n \left(\frac{\sigma_i}{\sigma_i^2 + \theta_i^2} \right) \mid 0_{n,m-n} \right] U^t (y_* + \varepsilon) \quad (3.52)$$

Then it follows that the value of the expectation is:

$$E_\varepsilon(\hat{w}) = V D_n \left(\frac{\sigma_i^2}{\sigma_i^2 + \theta_i^2} \right) V^t w_* \quad (3.53)$$

and the bias is:

$$b = E_\varepsilon(\hat{w}) - w_* = V D_n \left(-\frac{\theta_i^2}{\sigma_i^2 + \theta_i^2} \right) V^t w_* \quad (3.54)$$

The quantity:

$$\hat{w} - E_\varepsilon(\hat{w}) = V \left[D_n \left(\frac{\sigma_i}{\sigma_i^2 + \theta_i^2} \right) \mid 0_{n,m-n} \right] U^t \varepsilon \quad (3.55)$$

enables to calculate the variance $var(\hat{w}) = E_\varepsilon [(\hat{w} - E_\varepsilon(\hat{w}))(\hat{w} - E_\varepsilon(\hat{w}))^t]$ as:

$$var(\hat{w}) = \sigma_y^2 V D_n \left(\frac{\sigma_i^2}{(\sigma_i^2 + \theta_i^2)^2} \right) V^t \quad (3.56)$$

The notion of oracular regularization can now be introduced. Oracular regularization is the regularized solution that, based on the exact solution, gives the best estimator for w_* in expectation. That is, the oracular regularizer matrix T minimizes:

$$E_\varepsilon(\|y - y_*\|^2) = E_\varepsilon(\|X(\hat{w} - w_*)\|^2) \quad (3.57)$$

The central result can now be reported:

Theorem 3.2.1. *Given a sample set X and the functional (3.43) then a sufficient condition to obtain the best solution \hat{w} , for the measure $E_\varepsilon(\|(\hat{w} - w_*)\|^2)$ and $\hat{T} = TV^t$, is that the matrix T^2 has each diagonal element as:*

$$(\theta_i^{orac})^2 = \frac{\sigma_y^2}{\langle w_*, v_i \rangle^2} \quad (3.58)$$

Proof. See Appendix.

On this basis, then, oracular regularization needs, as shown in the proof, n independent terms θ_i^2 ; the choice $\theta_i^2 = \alpha \forall i$ which is the most popular in Tikhonov regularization cannot yield an oracularly regularized closed form solution.

3.2.1.1 Oracular Variance Estimator

Further developements of this generalized setting are based on the computation of an ideal variance estimator. The variance σ_y^2 of y by definition is:

$$\sigma_y^2 = \frac{1}{m} E_\varepsilon \{ (y - Xw_*)^t (y - Xw_*) \} \quad (3.59)$$

It is not difficult to show that the previous formula can be put in terms of following sum:

$$m\sigma_y^2 = e_1 + e_2 + e_3 \quad (3.60)$$

where the three terms are:

$$e_1 = E_\varepsilon \{ \|y - X\hat{w}\|^2 \} \quad (3.61)$$

$$e_2 = 2E_\varepsilon \{ (y - X\hat{w})^t X(\hat{w} - w_*) \} \quad (3.62)$$

$$e_3 = E_\varepsilon \{ (\hat{w} - w_*)^t X^t X(\hat{w} - w_*) \} \quad (3.63)$$

Working out the above decomposition leads to obtain the following result: the oracular unbiased variance estimator s^2 is given by:

$$s^2 = \frac{\|y - X\hat{w}\|^2}{m - r + \sum_{i=1}^r \frac{(\theta_i^{orac})^2}{\sigma_i^2 + (\theta_i^{orac})^2}} \quad (3.64)$$

Proof. See Appendix.

It should be stressed that this result follows from a rigorous evaluation and does not need the definition of degrees of freedom. It is now possible to define the degrees of freedom τ as:

$$\tau = m - r + \sum_{i=1}^r \frac{\theta_i^2}{\sigma_i^2 + \theta_i^2} \quad (3.65)$$

Thus one can for instance use the Generalized Cross Validation method to select the best model w :

$$GCV = \frac{\|X\hat{w} - y\|^2}{\tau^2} \quad (3.66)$$

Note that when $\hat{T} = \sqrt{\alpha}I_{nn}$ then the usual degree of freedom and thus usual GCV score are obtained.

3.2.2 On Learning

In this section the links between learning theory and oracularly regularized Tikhonov will be studied. Indeed it is well known the Tikhonov regularization can be used to address learning problems [40] such as regression or classification. From the machine learning perspective one can define a linear *decision function* given by $f(x) = wx$ which maps patterns x to a scalar $f(x)$. In the learning setting the matrix \hat{T} can be interpreted as a set of points where the decision function should be zero; this is due to the regularizer $\|\hat{T}w\|^2$ which promotes solutions $f(x)$ such that $f(\hat{T}) = \hat{T}w = 0$. We call this set of vectors \hat{T} , *Agnostic Vectors*. In particular when oracular regularization is applied, this set of points is *the best set of points* for Tikhonov regularization; for this reason one can call this set *Oracular Agnostic Vectors*.

These vectors allow one with a nice link with the concept of *Universum* introduced in [7]. In [7] it is shown that adding *non-examples* to the learning process can lead to better generalization; these samples are called *Universum* samples and lead to a functional called *Universum SVM* or U-SVM. An open question in that work is: *What is a good Universum for learning?* In the work [91] it is shown that a good Universum for U-LSSVM for classification is that in between the positive and negative clouds of the two classes: this result says that in the margin the decision function should be agnostic.

Looking at functional (3.43) one can observe that matrix \hat{T} could be interpreted as a samples set, i.e. a *Universum*. Oracular properties of Tikhonov regularization show that \hat{T} must have a particular structure in order to get in expectation the best result. This means that for Tikhonov regularization one knows how many and which are the samples of an ideal *Universum* of points. These points are equal to n and depend both on X and on the noise level σ_y .

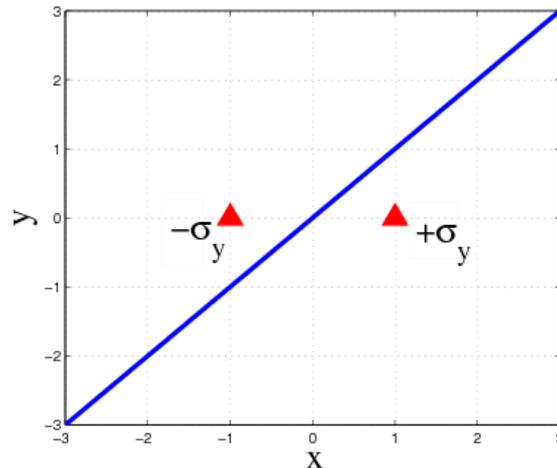


Figure 3.12: Agnostic Oracular Vectors for the univariate case and $w_* = 1$

In the monodimensional case an intuition of the behaviour of this points can be given.

Suppose the function to be learned is $g(x) = x$; then in this case $w_* = 1$. For uniformly sampled points one gets that V matrix of the SVD of X is the constant 1. Being in a monodimensional domain, then \hat{T} is a scalar term whose oracular value is:

$$\hat{T} = \pm \frac{\sigma_y}{\langle w_*, v \rangle} = \pm \sigma_y \quad (3.67)$$

Irrespective of the sign, which is influential owing to the symmetry, σ_y is the point in which the decision function should be agnostic, $f(\sigma_y) = \sigma_y w = 0$. Figure F1 depicts the situation: in the absence of noise, the oracular agnostic point is at $(0, 0)$; this is not a real constraint because the model Xw must pass through the origin by definition. In the presence of noise the constraint becomes active and try to force the regression function to pass in $(\sigma_y, 0)$: in other words, the function becomes more *smooth* and $\|w\|^2$ gets smaller when trying to fit $(\sigma_y, 0)$. Consistently, when the agnostic point is at infinity, the noise is infinite and w gets the zero value.

In the general case the matrix \hat{T} contains the weighted columns of the matrix V of SVD.

3.2.2.1 Oracular Linear Kernel

The link between oracular regularization and kernel regularized least square is now investigated. Generally speaking, the \hat{T} term induces a Reproducing Kernel Hilbert Space \mathcal{H} . This is true because the norm $w^t \hat{T}^2 w$ is a weighted norm of w , and \hat{T}^2 is a positive definite matrix. In terms of RKHS the functional with quadratic loss and decision function $f \in \mathcal{H}$ is:

$$\|y - f\|^2 + \|f\|_{\mathcal{H}}^2 \quad (3.68)$$

It is known [45] that the previous equation can be re-written by giving an explicit regularization operator P :

$$\|y - f\|^2 + \|Pf\|^2 \quad (3.69)$$

where now the norm is a classical L_2 norm. In the linear case $f(X) = Xw$ and so the functional becomes:

$$\|y - Xw\|^2 + \|PXw\|^2 \quad (3.70)$$

In this equation it is immediate to identify that PX is the analogous of \hat{T} . From [45] it is known that the kernel matrix associated to the space \mathcal{H} can be obtained by $K = (P^t P)^{-1}$. In the following, for simplicity it will be assumed that P is one-to-one operator and that X is square and full rank. This leads to $P = \hat{T}X^{-1}$. Now the kernel matrix K associated to the space \mathcal{H} can be explicitly computed (see Appendix for the proof):

$$K = UD_m \begin{pmatrix} \frac{\sigma_i^2}{\theta_i^2} \end{pmatrix} U^t \quad (3.71)$$

On this basis, the problem on kernel space can be formulated as:

$$\min_{\psi} \|y - K\psi\|^2 + \psi^t K \psi \quad (3.72)$$

where ψ is the variable on the kernel space. The minimizing value $\hat{\psi}$ of ψ is:

$$\hat{\psi} = (K + I_m)^{-1} y = UD_m \begin{pmatrix} \frac{\theta_i^2}{\theta_i^2 + \sigma_i^2} \end{pmatrix} U^t y \quad (3.73)$$

Then, by using (3.52) one gets:

$$\hat{w} = VD_m \left(\frac{\sigma_i}{\sigma_i^2 + \theta_i^2} \right) U^t y \quad (3.74)$$

It is easy to verify that the relation between w and ψ is:

$$w = V\Sigma^t T^{-2} U^t \psi \quad (3.75)$$

In the case $T = I$ as for classical Tikhonov regularization the relation between ψ and w reduces to the well known result [53]:

$$w = X^t \psi \quad (3.76)$$

With analogy to the previous oracular analysis the notion of *oracular linear kernel* can be introduced. This kernel is computed by plugging oracular regularization (3.58) on K . It follows that the oracular kernel and the oracular solution are:

$$K^{orac} = UD_m \left(\frac{\sigma_i^2 < w_*, v_i >^2}{\sigma_y^2} \right) U^t \quad (3.77)$$

$$\psi^{orac} = UD_m \left(\frac{\sigma_y^2}{\sigma_y^2 + \sigma_i^2 < w_*, v_i >^2} \right) U^t y \quad (3.78)$$

Then the *oracular prediction* on training data is:

$$K^{orac} \psi^{orac} = UD_m \left(\frac{\sigma_i^2 < w_*, v_i >^2}{\sigma_y^2 + \sigma_i^2 < w_*, v_i >^2} \right) U^t y \quad (3.79)$$

It is interesting to observe that when the noise level $\sigma_y^2 = 0$ the oracular prediction is y . This is consistent with the fact that one is observing the true noiseless data $y = y_*$.

3.2.3 On Shrinking

The generalized form of Tikhonov regularization can be effectively used in the statistical domain of shrinking; in this setting one is requested to define the *optimal* shrinking coefficient such that a statistical estimate (i.e. the mean) is improved in expectation. Now it is shown how shrinking, and in particular the d -means problem [92] can beneficiate from the oracular form of Tikhonov regularization.

The first point that can be discussed is that the regularized mean problem [4] is a particular instance of Tikhonov regularization. The regularized mean problem poses the following issue: given a gaussian distribution $N(\mu, \sigma)$ from which n samples are given, deduce the true mean value. The usual sample mean formula can be used, indeed this is the classical maximum likelihood solution; however the work in [4] showed that regularization-inspired solutions be used.

In this context, the regularized mean problem can be seen as a special case of regression where only a constant term (the mean) is fitted:

- A Gaussian distribution $N(\mu, \sigma)$ produces n samples x_i
- $X = I_{nn}$
- w is a vector with all equal entries ξ
- The y vector of regression is mapped to the samples $x_i = y_i$

Then one gets:

$$\bar{x} = \arg \min_w \sum_i^n (x_i - w_i)^2 + \|\hat{T}w\|^2 \quad (3.80)$$

In this case the optimal vector w_* is the vector made all of the same scalar μ . Moreover being $X = I_{nn}$, $\hat{T} = T$. Finally, the oracular regularizer is:

$$(\theta_i^{orac})^2 = \frac{\sigma_y^2}{\langle w_*, v_i \rangle^2} = \frac{\sigma^2}{n\mu^2} \quad (3.81)$$

and all thetas are equal.

If one set $\theta_{(n)}^2 = n\theta^2$ a possible rewriting of (3.80) is :

$$\bar{x} = \arg \min_{\xi} \sum_i^n (x_i - \xi)^2 + \theta_{(n)}^2 \xi^2 \quad (3.82)$$

That is identical to the equation proposed in [4]. One gets that: $(\theta_{(n)}^{orac})^2 = \sigma^2/\mu^2$ that is identical to the result obtained in [4]. Thus the notion of oracular regularization proposed here is consistent with that in [4].

3.2.3.1 d -means Problem and James-Stein estimator

In the d -means problem one has d gaussians $N(\mu_i, \sigma)$ and from each only one sample is available. The goal is to estimate the vector μ of means $\mu_i, \forall i$. The obvious solution is to estimate each mean by the only available sample of its corresponding distribution, thus $\hat{w} = x$; this solution is not the best in terms of the expected cost $E_\varepsilon \|\mu - \hat{w}\|^2$; indeed the classical work of James-Stein showed that a better estimator than that given by $\hat{w} = x$ can be obtained by shrinking the samples x .

Mapping this setting to Tikhonv leads to the following specifications:

- d gaussian distributions $N(\mu_i, \sigma)$, and each produces a sample $x_i, i = 1, \dots, d$
- $X = I_{dd}$
- The y vector of regression is mapped to the samples $x_i = y_i$

Given these premises the regularized d -means problem becomes:

$$\bar{x} = \arg \min_w \sum_{i=1}^d (x_i - w_i)^2 + \|\hat{T}w\|^2 \quad (3.83)$$

The oracular regularizers for such a problem are:

$$(\theta_i^{orac})^2 = \frac{\sigma^2}{\langle w_*, v_i \rangle^2} = \frac{\sigma^2}{\mu_i^2} \quad (3.84)$$

From the cost functional of the d -means it is easy to show that $\bar{x}_i = \left(\frac{1}{1+\theta_i^2}\right) x_i$. Using oracular regularization one gets:

$$\bar{x}_i = \left(\frac{\mu_i^2}{\mu_i^2 + \sigma^2}\right) x_i = \left(1 - \frac{\sigma^2}{\mu_i^2 + \sigma^2}\right) x_i \quad (3.85)$$

A first surprising aspect regards the case $d = 1$. In this case it is not true that the best result is given by the unique sample x , instead its regularized version is better:

$$\bar{x} = \left(1 - \frac{\sigma^2}{\mu^2 + \sigma^2}\right) x \quad (3.86)$$

This produces a sort paradox for which, given one sample it is still appropriate to shrink it!

One should observe that (3.85) can be also written as:

$$\bar{x}_i = \left(\frac{1}{1 + NSR_i} \right) x_i \quad (3.87)$$

where $NSR_i = \sigma^2/\mu_i^2$ indicates the noise to signal ratio. When written in this way equation (3.85) resembles the structure of Wiener filter where noise to signal ratio is the regularizing factor (in the following an entire section is dedicated to filters and generalized Tikhonov).

A second interesting aspect of these regularizers emerges when compared to the regularizer of James-Stein:

$$\bar{x}^{JS} = \left(1 - \frac{\sigma^2}{\|x\|^2/(d-2)} \right) x \quad (3.88)$$

The James-Stein regularizer depends on the global norm $\|x\|$; this means that each \bar{x}_i^{JS} depends on a global quantity. Differently the oracular regularizer shows that the best estimator can be obtained when each x_i is regularized *independently* of each other and no global information is used to regularize each single component of \bar{x}_i . This, in turn, means that, from the oracular point of view, the so-called Stein paradox does not take place because each component of the best estimator (i.e. oracular) is independent from the others.

Then the oracular regularization solution, which has independent oracular components, dominates the James-Stein solution, where each $\mu_i^2 + \sigma^2$ term is replaced by the global quantity $\frac{\sum_{i=1}^d (\mu_i + \sigma \varepsilon_i)^2}{d-2}$.

Owing to the structure of the oracular regularizers one can try to give regularizers/shrinkers different from that given by James-Stein.

In particular we propose two oracularly-inspired regularizers: the first maintains the independent component feature of oracular regularization:

$$\bar{x}_i^{(1)} = \left(1 - \frac{\sigma^2}{x_i^2 + \sigma^2} \right) x_i \quad (3.89)$$

The second one uses the intuition of James-Stein of computing a global quantity: roughly speaking one estimates each μ_i by the unique *crossed* sample

mean $\tilde{x}_0 = \sum_{i=1}^n x_i/n$ which mix up samples from the different distributions $N(\mu_i, \sigma)$:

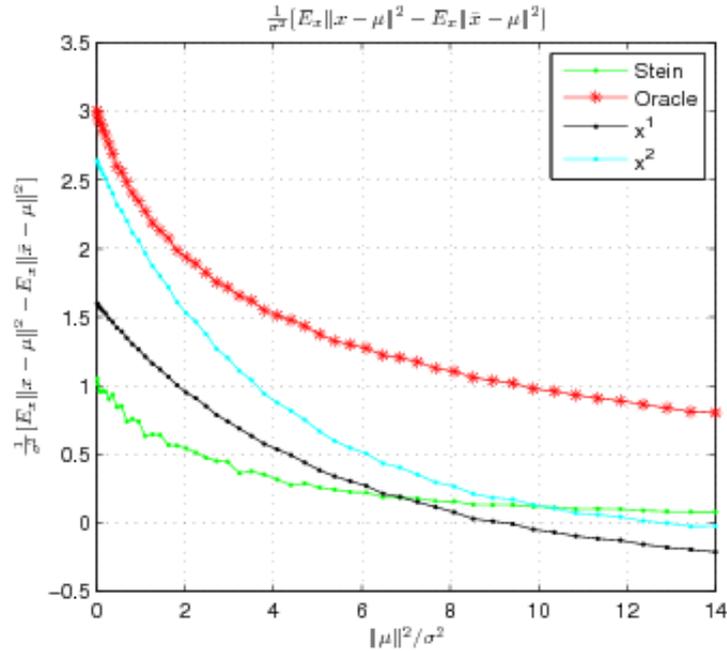
$$\bar{x}_i^{(2)} = \left(1 - \frac{\sigma^2}{\tilde{x}_0^2 + \sigma^2}\right) x_i \quad (3.90)$$

The intuition behind this regularizer is that if the means components of the mean vector μ are all of the same sign then the crossed sample estimator is a reasonable estimator of the true means, thus can be used as a proxy of μ in the regularizer. We analyzed the case $d = 3$ and the range $1 \leq \|\mu\|^2/\sigma^2 \leq 14$; the i -th means vector was defined as $\mu_i = i/N * \{1, 2, 3\}$ where $i \in [1, N]$ and $N = 50$. The results are evaluated by computing (via Montecarlo approximation) the expected normalized gap $(1/\sigma^2)[E\|\mu - x\|^2 - E\|\mu - \bar{x}\|^2]$ where \bar{x} is either $\bar{x}^{(1)}$ or $\bar{x}^{(2)}$. In figure 3.13 the black line indicates $\bar{x}^{(1)}$ the cyan line indicates $\bar{x}^{(2)}$, green line indicates Stein solution, and red line indicates oracular solution; the higher the value the more effective is the regularization strategy; a negative value of the gain indicates that the unregularized solution is better. One should also note that the proposed regularizer can be also used for $d = 2, 1$; results in this case show that $\bar{x}^{(2)}$ still gives a useful solution but in a narrower range of signal to noise ratio.

Results show that the \bar{x}_2 regularizer is significantly better than Stein until $\|\mu\|^2/\sigma_y^2 = 10$; over this value Stein is more effective and $\bar{x}^{(2)}$ is even a bit less effective than non regularizing. However, in the interesting region $\|\mu\|^2/\sigma^2 \leq 10$, where noise is significant, the proposed regularizer shows a significantly better behaviour than Stein and gives nearly oracular results for $\|\mu\|^2/\sigma^2 \leq 2$, that is, when noise gets significant with respect to the signal. Analogous results hold for different values of d ; the obtained experimental results showed that, knowing the closed form solution of the regularizers, lead to a very effective regularizer that significantly improve the historical James-Stein result; interestingly the closed form formula of $\bar{x}^{(2)}$ is simple as that of Stein.

3.2.3.2 Shrinking Extensions to Regression

James-Stein machinery can be extended to regression; in particular Copas in [90] gave the closed form regularization *a la* James-Stein for regression;

Figure 3.13: Regularizers for the d -means problem

now the aim is comparing Copas results to oracular solutions.

Given the data matrix X Copas defines the matrix $V_c = \frac{1}{n}X^tX$. Then one defines M as an orthogonalizing matrix for V_c that is M obeys $MV_c^{-1}M^t = I_{nn}$. Copas does not give a operative definition of M ; a definition based on SVD is particularly convenient for the subsequent manipulations. Using the SVD for X it is easy to show that:

$$V_c = \frac{1}{m}V\Sigma^t\Sigma V^t \quad (3.91)$$

from which it follows that a suitable definition of M is:

$$M = \frac{1}{\sqrt{m}}\Delta^tV^t \quad (3.92)$$

where $\Delta = \Sigma^t\Sigma$. M fulfils the orthogonalizing property $MV_c^{-1}M^t = I_n$ Given these premises and denoting by \hat{w}_0 the unregularized solution, Copas shows that a Stein-like shrinker can be used to compute the regularized solution :

$$\hat{w}_{cp} = \left(1 - \frac{(n-2)\sigma_y^2}{n\hat{w}_0^t M^t M \hat{w}_0}\right) \hat{w}_0 \quad (3.93)$$

which in terms of the SVD elements becomes:

$$\hat{w}_{cp} = \left(1 - \frac{(n-2)\sigma_y^2}{\hat{w}_0^t V \Delta^2 V^t \hat{w}_0}\right) \hat{w}_0 \quad (3.94)$$

It is interesting to study the one dimensional case for Copas regularization. Starting from (3.94) one gets the univariate Copas regularizer as:

$$\hat{w}_{cp} = \left(1 + \frac{\sigma_y^2}{\hat{w}_0^2 v^2 \sigma_1^2}\right) \hat{w}_0 \quad (3.95)$$

where σ_1 is the only singular value. It can be rewritten as:

$$\hat{w}_{cp} = \left(\frac{\hat{w}_0^2 v^2 \sigma_1^2 + \sigma_y^2}{\hat{w}_0^2 v^2 \sigma_1^2}\right) \hat{w}_0 \quad (3.96)$$

An interesting property of this last shrinker follows considering the oracular regularization in the mono-dimensional case. In this case, $X^t X$ is a scalar, thus the oracular solution \hat{w}^{orac} is simply:

$$\hat{w}^{orac} = \left(\frac{1}{X^t X + (\theta^{orac})^2}\right) X^t y = \left(\frac{X^t X}{X^t X + (\theta^{orac})^2}\right) \hat{w}_0 \quad (3.97)$$

Recalling that $X^t X = \sigma_1^2 v^2$ and $(\theta_i^{orac})^2 = \frac{\sigma_y^2}{(w_*)^2 v^2}$ one obtains:

$$\hat{w}^{orac} = \left(\frac{\sigma_1^2 v^4 w_*^2}{\sigma_1^2 v^4 w_*^2 + \sigma_y^2}\right) \hat{w}_0 \quad (3.98)$$

When x is monodimensional then V matrix collapse to the value 1. Thus equations (3.98) and (3.96) look similar: replacing w_* with \hat{w}_0 in (3.98) than one obtains that the oracular shrinker is the reciprocal of the Copas shrinker. This suggests that Copas shrinking has a complementary behaviour with respect to shrinking induced by a oracular regularizer where \hat{w}_* is substituted by \hat{w}_0 . These interesting properties hold for a monodimensional domain; when the domain is multidimensional the equivalence between regularization and shrinking for regression is no more true; indeed shrinking only acts with a shrinking coefficient λ such that $\hat{w}_{cp} = \lambda \hat{w}_0$, instead in Tikhonov regularization the relation between \hat{w} and \hat{w}_0 is $\hat{w} = L \hat{w}_0$ where $L = (X^t X + \hat{T}^2)^{-1} X^t X$, thus shrinking for regression is less powerful and can be considered as a form of *weak regularization*.

3.2.4 On Filtering

The aim of this section is to show that Tikhonov regularization has a tight relation with filtering; in particular it will be shown that oracular Tikhonov corresponds to the Wiener filter; hence it is shown that *filtering is a special case of regularization/learning where the operator X is a circular convolution operator*. Further, owing to the sensitivity of Tikhonov regularization to outliers, an alternative filter to Wiener's is proposed.

The connections between regularization and filtering can be shown considering a deconvolution problem with noise. Given a filter function in time $x(t)$ and a signal $w_*(t)$, the noisy convolved signal $y(t)$ is defined as:

$$y(t) = x(t) \star w_*(t) + \varepsilon(t) \quad (3.99)$$

where $\varepsilon(t)$ is white Gaussian noise of known variance σ^2 . Assume that $x(t), y(t), w_*(t)$ conceptually correspond to the vectors x, y, w_* defined in the previous section where now each component of each vector has the meaning of value at a time instant and all vectors are of the same length n . From now on the symbol $\mathcal{F}(\cdot)$ will indicate the Fourier transform and $\mathcal{F}^{-1}(\cdot)$ will indicate the inverse Fourier transform; $w(t)$ indicates the filtered signal, additionally one has the following definitions:

1. $X(f) = \mathcal{F}(x(t))$
2. $W(f) = \mathcal{F}(w(t))$
3. $W_*(f) = \mathcal{F}(w_*(t))$
4. $V(f) = \mathcal{F}(\varepsilon(t))$
5. $Y(f) = \mathcal{F}(y(t))$

Given a filter $G(f)$ the operation of filtering in frequency is obtained by:

$$W(f) = G(f)Y(f) \quad (3.100)$$

It is well known that the ideal deconvolutive filter using as quality measure $E_\varepsilon \|w(t) - w_*(t)\|^2$ (by Parseval theorem the same holds in the frequency domain) is the Wiener filter $G(f)$ defined in frequency by:

$$G(f) = \left(\frac{X^*(f)}{|X(f)|^2 + NSR(f)} \right) \quad (3.101)$$

where $NSR(f)$ is the Noise Signal ratio, that in the homoskedastic case is:

$$NSR(f) = \frac{\sigma^2 n}{|W_*(f)|^2} \quad (3.102)$$

In order to show the equivalence of a particular case of Tikhonov regularization with a deconvolution problem, the Fourier matrix F_n of size $n \times n$ must be introduced. Let $\omega = e^{-2\pi j/n}$ where j is the imaginary unit. The Fourier matrix F_n is defined by:

$$F_n = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{pmatrix} \quad (3.103)$$

Depending on the definition of F_n a term $1/\sqrt{n}$ is sometimes introduced as a rescaling of the matrix; in this work, however, the definition (3.103) without rescaling factor will be used. Fourier matrix has several nice features:

- $F_n^t = nF_n^{-1}$
- $W(f) = \mathcal{F}(w(t)) = F_n w$
- $w(t) = \mathcal{F}^{-1}(W(f)) = \frac{1}{n} F_n^t w$
- If X is a circulant matrix then it is known that Xw has the same effect of the circular convolution $x(t) \star w(t)$. In this particular case then the Fourier matrix F_n diagonalizes X as:

$$X = F_n^{-1} S F_n \quad (3.104)$$

where S is the eigenvalues diagonal matrix. The eigenvalues follow from the Fourier transform of the first column of X , that is:

$$S = \text{diag}(F_n X(\cdot, 1)) \quad (3.105)$$

When analyzing the general case of Tikhonov regularization it was convenient to use as rotation matrix R the matrix V^t provided by SVD. In this case X is a particular matrix, a circulant one, and one can adopt as rotation matrix the Fourier matrix F_n . Therefore one can define the matrix \hat{T} as:

$$\hat{T} = T_{\mathcal{F}} F_n \quad (3.106)$$

where $T_{\mathcal{F}}$ is a diagonal matrix and has the same role of T when dealing with Tikhonov regularization in the general case; thus $\hat{T}^2 = F_n^t T_{\mathcal{F}}^2 F_n$. The solution to the generalized Tikhonov functional is given by solving:

$$(X^* X + F_n^t T_{\mathcal{F}}^2 F_n) w = X^* y \quad (3.107)$$

Using the fact that X is a convolution operator one gets:

$$((F_n^{-1} S F_n)^* F_n^{-1} S F_n + F_n^t T_{\mathcal{F}}^2 F_n) w = F_n^* S^* (F_n^{-1})^* y \quad (3.108)$$

Racalling that $F_n F_n^t = n I_{nn}$ then:

$$\left(\frac{1}{n} F_n^* S^* S F_n + F_n^* T_{\mathcal{F}}^2 F_n\right) w = \frac{1}{n} F_n^* S^* F_n y \quad (3.109)$$

Collecting and simplyfing one gets:

$$(S^* S + n T_{\mathcal{F}}^2) F_n w = S^* F_n y \quad (3.110)$$

by using the property that $X(f) = F_n x$ one gets:

$$(S^* S + n T_{\mathcal{F}}^2) W(f) = S^* Y(f) \quad (3.111)$$

It follows:

$$W(f) = S^* (S^* S + n T_{\mathcal{F}}^2)^{-1} Y(f) \quad (3.112)$$

Noting that both S^* and $(S^* S + n T_{\mathcal{F}}^2)$ are diagonal, the inversion collapses to point-wise division. Recalling that: $X(\cdot, 1)$ is the zero padded signal $x(t)$ and S matrix embeds the Fourier transform of $X(\cdot, 1)$ in its diagonal, then one gets:

$$W(f) = \frac{X^*(f)}{(|X(f)|^2 + n d_{\mathcal{F}}^2)} Y(f) \quad (3.113)$$

where $d_{\mathcal{F}}$ is the diagonal in $T_{\mathcal{F}}$.

The obtained result is a Wiener-type filter where $NSR(f) = n d_{\mathcal{F}}^2$.

Now the term $d_{\mathcal{F}}^2$ is studied and its oracular value is derived.

One can build a parallel between generalized Tikhonov results and the current special case: in the general case X has the SVD decomposition $X = U\Sigma V^t$; if X additionally is a circular convolutive matrix, as in this case, then $X = F_n^{-1}SF_n$; hence F_n and F_n^t , in this context, play the role of V^t and V respectively; all the given proves for oracular optimality and the general case can be repeated by using $X = F_n^{-1}SF_n$ instead of $X = U\Sigma V^t$. Then one gets that in the Fourier domain the oracular regularizers are:

$$\hat{\theta}_i^2 = \frac{\sigma_y^2}{|w_*^t F_n^t(i)|^2} \quad (3.114)$$

where $F_n^t(i)$ is the i -th column of F_n^t . Note that $\hat{\theta}_i^2$ are the elements on the diagonal of $T_{\mathcal{F}}^2$ and thus they form the vector $d_{\mathcal{F}}^2$.

The meaning of the scalar elements $|w_*^t F_n^t(i)|^2$ can be evidenced as follows:

$$|w_*^t F_n^t(i)|^2 = |F_n(i)w_*|^2 = |W_*(f_i)|^2 \quad (3.115)$$

where $F_n(i)w_*$ is the i -th component of the Fourier transform $F_n w_*$ and $F_n(i)$ indicates the i -th row of F_n . This means that:

$$\hat{\theta}_i^2 = \frac{\sigma_y^2}{|W_*(f_i)|^2} \quad (3.116)$$

Plugging this equation into the Tikhonov solution (3.113) and letting $f = [f_1, \dots, f_n]$ the vector of the frequencies one gets:

$$W(f) = \frac{X * (f)}{|X(f)|^2 + \frac{n\sigma_y^2}{|W_*(f)|^2}} Y(f) \quad (3.117)$$

That is exactly the Wiener filter.

3.2.5 Non linear extensions

On the basis of the equivalence Tikhonov-Wiener one can observe that some of the non desirable aspects of Tikhonov regularization reflect the Wiener filter. For instance, a severe drawback of Tikhonov regularization is that, due to the square loss, Tikhonov is quite sensitive to *outliers*, where outliers are the patterns presenting anomalies such as spiking noise for signals. In the

presence of outliers, Wiener filter can give unsatisfactory results. One can try to improve the results by modifying Tikhonov loss function. An example of modified functional is:

$$\min_w \sum_{i=1}^n \hat{\mathcal{L}}(r_i) + \alpha \|w\|^2 \quad (3.118)$$

where γ is a parameter, $r_i = y_i - w \cdot x_i$ and:

$$\hat{\mathcal{L}}(r_i) = |r_i| - e^{-\gamma r_i^2} \quad (3.119)$$

The loss in the previous equation rejects well outliers because weights *big* errors much less than the Tikhonov quadratic loss; other possible examples are norm one loss, Huber loss, bi quadratic etc..

The functional (3.118) can be minimized by employing Iteratively Reweighted Least Squares (IRLS): as usual in IRLS [93] W is the diagonal weights matrix whose weights are the derivatives on r_i of the loss function $\hat{\mathcal{L}}$ divided by the error r_i . Thus one gets:

$$W(i, i) = \frac{\text{sign}(r_i)}{r_i} + 2\gamma e^{-\gamma r_i^2} \quad (3.120)$$

The steps employed by the algorithm are the following:

1. Set as initial solution w the solution obtained by Wiener filter, that corresponds to unitary weights.
2. Set $w_{old} = w$, compute the residuals $r = Wy - WXw$
3. Build the diagonal matrix W whose i -th diagonal element as per equation (3.120) where
4. Update the solution by solving $(X^tWX + \alpha I)w = X^tWy$
5. Stop if $\|w_{old} - w\|^2 \leq \tau$, where τ is a tolerance or a max number of iterations is reached. Else goto 2

This kind of approach can be used in a deconvolution with noise plus outliers problem. To simulate the presence of outliers in a physically plausible

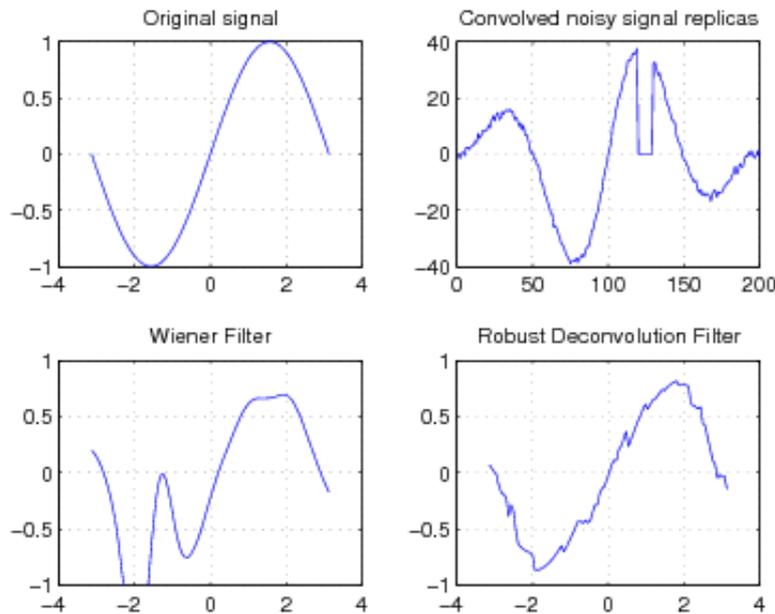


Figure 3.14: Wiener vs Robust filtering

way one can imagine a device which during the convolution process sometimes stop working in non predictable way. In practical terms this means that a convolved signal has outliers where outliers are points where no signal is present; in terms of images this in turn could mean that camera has some broken CCD pixels, in other terms pixels that produce black points.

In the monodimensional domain, as an example, suppose that $t \in [-\pi, +\pi]$, $w(t) = \sin(t)$ and $x(t) = \cos(t)$. Suppose that $y(t) = x(t) \star w(t) + \varepsilon(t)$, where $\varepsilon(t)$ is white gaussian noise distributed as $N(0, 1)$. To simulate the presence of phisically plausible outliers, at certain time instance some data of $y(t)$ is nullified thus simulating a device badly working. The filter in (3.118) has two parameters, one is α and the other is γ ; the first is to n (that is the size of $x(t)$) and γ is set to 1. Figure 3.14 show the results obtained with the filter (3.118) and the ideal Wiener where NSR is estimated at each frequency; in other words robust filtering is using one regularization term, instead Wiener filter is using multiple regularization terms.

Robust filtering is more effective than Wiener filter in such situation. Called $w_W(t)$ the Wiener reconstructed signal and $w_r(t)$ the robustly reconstructed

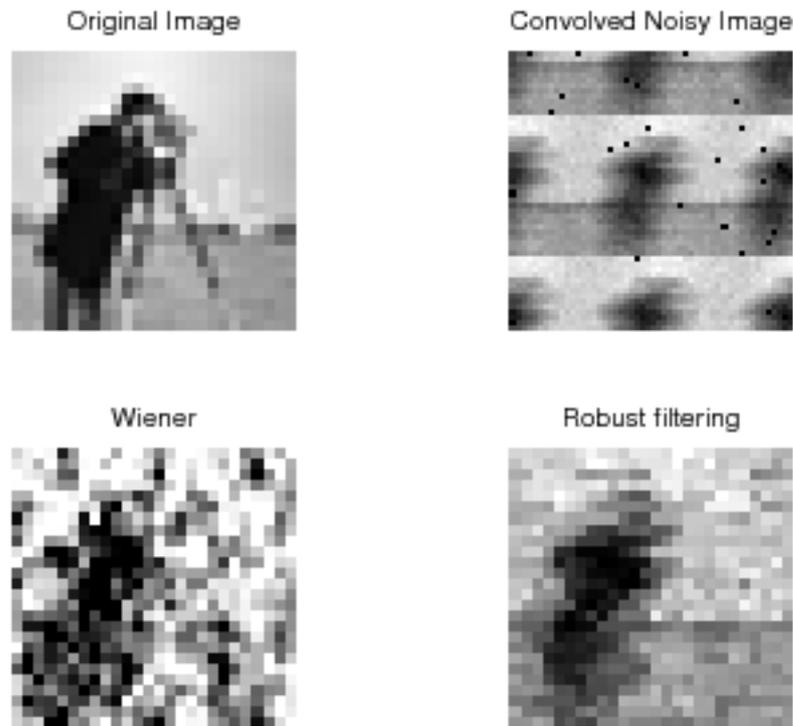


Figure 3.15: Wiener vs Robust filtering 2D

signal one gets that $\|w_* - w_W(t)\| = 3.6923$ and $\|w_* - w_r(t)\| = 1.8744$.

The same reasoning can be carried for images. In this case the images are vectorized and everything is recast as a one dimensional problem; now $\alpha = 1e - 5 * n$ where n is the size of the vectorized image (included zero padding), $\gamma = 1$ and the convolution matrix is a motion blur. Also in this case the reconstruction is more robust than the Wiener filter (see figure 3.15): $\|w_* - w_W(t)\| = 844$ and $\|w_* - w_r(t)\| = 348$.

This brief example showed how on the basis of the equivalence between learning regularization and filtering than one is able to bring ideas from a field to another; for instance, as shown here, an easy way to cope with Tikhonov sensitivity to outliers is to exponentially weight the errors; this approach would have been impossible while directly working in frequency; indeed one can observe that IRLS cannot be carried in frequency but must be performed in

X matrix	$\mathcal{L}(y, X, w)$	$\Omega(w)$	Problem/Algorithm
Any	$\ Xw - y\ _2^2$	$\ Tw\ _2^2$	Regularization/Tikhonov Learning
Any	$(Xw - y)_\epsilon$	$\ Tw\ _2^2$	SVM Learning
Any	$\sum_i^n \log(1 + \exp(-y_i(w \cdot x_i)))$	$\ w\ _2^2$	Logistic Regression
Identity	$\ Xw - y\ _2^2$	$\ Tw\ _2^2$	Shrinking
Convolutive	$\ Xw - y\ _2^2$	$\ Tw\ _2^2$	Deconvolution
Measurement matrix	$\ Xw - y\ _1$	$\ w\ _1$	Compressed sensing
Any	$\ Xw - y\ _2^2$	$\ w\ _1$	Lasso

Table 3.1: Regularization, Learning, Filtering, and Shrinking functional form

time; whenever non linearity is used the frequency domain is lost.

3.3 Discussion and Conclusion

It has been shown how a generalized approach to Tikhonov allows to give insights on regularization itself but also in shrinking, on filtering and learning. Lastly it has been shown how a generalization of Tikhonov regularization immediately leads to an effective remedy against outliers; this is not the only possibility and other generalizations are possible. Owing to the equivalence of Tikhonov and Wiener one can further generalize Tikhonov functional to the form:

$$\hat{w} = \arg \min_w \mathcal{L}(y, X, w) + \alpha \Omega(w) \quad (3.121)$$

where $\mathcal{L}(y, X, w)$ indicates an arbitrary loss function, and $\Omega(w)$ is an arbitrary regularizer. Within this formulation several learning and filtering problems can be summarized depending on \mathcal{L} , $\Omega(w)$ and the meaning of X . Table 3.1 summarizes the different problems depending on the exposed factors.

In table 3.1 a *Measurement matrix* is matrix that mimicks the measurement of a signal; an example of such matrices are Random Projections matrices that are matrices that follows the Restricted Isometry Property, that is they are quasi-orthogonal (see [94]) and $(Xw - y)_\epsilon$ indicates the SVM ϵ -insensitive loss. Hence the same functional (3.121) allows to explain together a large number of algorithms and problems in different statistical, mathematical engineering domains.

The contamination among these generalized models can lead to generalized notions of shrinking and filtering and the reciprocal convergence of method-

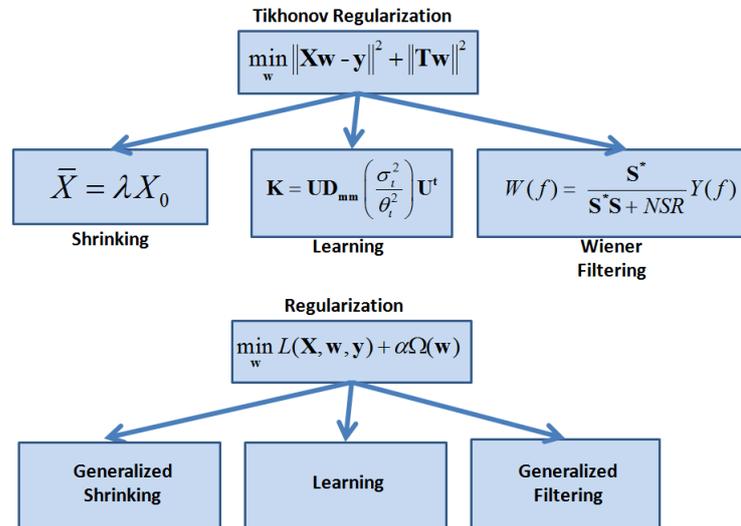


Figure 3.16: Tikhonov equivalence scheme and generalized learning, shrinking and filtering

ologies can be beneficial for all the fields; thus one can envision the general scheme in figure 3.16 where generalized notions are used.

Summarizing we showed how Tikhonov functional is able to explain different issues in different domains; moreover the central obtained result on oracular regularization made us able to give insights on shrinking and on the behaviour of James-Stein regularizer; in particular, we gave an effective non uniform improvement over James-Stein regularizer. Notably oracular properties showed that James-Stein paradox, at the optimum, vanishes. From the learning point of view a notion of oracular linear kernel was introduced, in addition in the filtering domain the complete adherence of Wiener filtering to generalized Tikhonov was established. Lastly a general view of the regularization, learning, shrinking and filtering problem is given in which disciplines contaminate each other; as exemplum of this contamination by modifying Tikhonov loss function a robust filter for deconvolution is proposed.

3.4 Biased Regularization for Controlling Capacity

Learning machines acquire knowledge by adjusting a model empirically in compliance with training data, hence the correct selection of the model parameters is a crucial issue. That choice is usually driven by predicting the generalization performances that are associated with the possible model settings. From a most general viewpoint, the bound to the generalization error, $R[f]$, results from the sum of two terms: the empirical error on the training sample, $R_{emp}[f]$, and a penalty term, $C_{\mathcal{F}}$, associated with the complexity:

$$R[f] = R_{emp}[f] + C_{\mathcal{F}} \quad (3.122)$$

In the case of Support Vector Machines for classification [3], the literature offers a variety of theoretical approaches to attain an (usually upper-bounded) estimate of the generalization error [3]. This proves especially useful in the presence of limited training samples, when classical techniques such as Cross-Validation [95] may waste samples because one has to exclude training patterns from the parameter adjustment. The Maximal-Discrepancy (MD) approach [26] is often effective in those cases for estimating the generalization penalty, $C_{\mathcal{F}}$. The computation of the latter quantity requires several, independent training processes on random label configurations. In most theoretical approaches, however, the resulting bounds may prove quite loose because the assumptions involved are, in general, strongly conservative. This typically leads to large values of the penalty terms, $C_{\mathcal{F}}$, of the bounds themselves.

Within that framework, this section introduces a novel model called VQSVM: this machine is an auxiliary machine that supports the computation of tighter penalty terms, $C_{\mathcal{F}}$ for SVM. The method sets two fixed references: the SVM solution of the classification problem with the original classes, and an unsupervised representation of the training patterns. In the computation of the penalty term for the basic classifier, any contribution is generated by a set of possible hypothesis that are constrained by these two references.

The unsupervised analysis of training data is based on Kernel Vector Quantization (VQ) for unsupervised clustering; hence an implicit *cluster hypothesis*

is made on data to get tighter bounds. This hypothesis is typical of semi-supervised learning approaches [96, 97]; in the present research (and also in [98, 77]) this hypothesis is used only on labeled data and does not involve the usual dichotomy between labeled and unlabeled data. Finally the MD approach is used as a tool to compute the penalty term.

Summarizing, VQSVM is an instance of Ivanov-like biased regularization, where the biasing is induced by the clustering process: the VQSVM methodology, however, features a wider general validity and could be used to enhance other learning machines.

Results shows that the method always improves on, or at least is equivalent to the classical Structural Risk Minimization paradigm [3], whose results reduce to a special (worst-)case of the VQSVM predictions.

At the same time, the convexity of the VQ-constrained problem formulation is proved; the (global) optimal solution is always achieved when one minimizes the VQSVM functional to compute the generalization bound. Finally, an efficient procedure to compute the penalty term under the unsupervised-constrained assumption is derived.

The method effectiveness is demonstrated on different, real-world testbeds: the NIST dataset [99], the Newsgroup-20 text-mining dataset [100], and several other UCI datasets [100]. The former two cases are significant because of the high-dimensional nature of the data space, the large size of the sample and the full compliance to the cluster hypothesis. Empirical results show that the VQSVM method succeeds in improving model selection whenever possible, and always yield tighter generalization bounds than those predicted by conventional MD methods.

3.4.1 Constraining SVM Capacity by Unsupervised Analysis

The VQSVM method is effective in the estimation process of the penalty term $C_{\mathcal{F}}$, which is estimated by solving a battery of SVM training problems. The main idea is to set the result of unsupervised Vector Quantization as a reference solution to constrain the capacity of each SVM in that battery.

This section shows that adding a suitable constraint to the SVM formulation leads to a refined model (VQSVM), which replaces the original SVM problem

setting in the computation of the penalty term on random target configurations. It is worth stressing that the VQSVM approach does not affect the first term, in the generalization estimate, which always depends on training data (n samples) and is computed by training a classical SVM on real classes. Instead, the proposed method contributes to the computation of the sample-based, target-independent evaluation of the penalty term; it does not affect the complexity of the classifier trained on real labels, but reduces the complexity of the several classifiers that are involved in the computation of the bounding term.

3.4.1.1 Unsupervised learning in the kernel space

The kernel-based unsupervised representation in the feature space follows a two-step process:

1. A classical unsupervised clustering (in fact, a dichotomy) of training patterns.
2. A Support Vector-based representation of the clustering results.

Step 1) The Kernel k -means algorithm (see 2.3.1) [78, 74] divides the projections, ϕ_i , of input data into two clusters, C_j , ($j = 0, 1$). The algorithm operates on distance values that are computed by using the kernel trick without the explicit coordinates of cluster centroids, Ψ_j . Let the “membership vector” $\mathbf{m} \in \{0, 1\}^n$ encode the partitioning of input patterns into the two clusters: $m_i = 0$ if $\phi_i \in C_0$ and $m_i = 1$ if $\phi_i \in C_1$, $i = 1, \dots, n$. Each prototype lies in the centroid of its associate partition, hence the membership vector \mathbf{m} determines the prototypes’ positions even though they are not stated explicitly.

Step 2) The pair of clusters obtained from step 1) supports an (artificial) classification problem in which the cluster membership of each pattern sets the provisional class of the pattern itself. In this respect, one cannot decide *a priori* which artificial label $\{+1, -1\}$ should be assigned to either cluster, thus one builds up an artificial training set, Z^+ , whose elements are labeled as: $Z^+ = \{(\mathbf{x}_i, y_i^{(KM)+}); i = 1, \dots, n; y_i^{(KM)+} = 2m_i - 1\}$. This set undergoes a conventional SVM training process. The resulting hyper-plane, $\mathbf{w}^{(KM)+}$, is given

by:

$$\mathbf{w}^{(KM)+} = \sum_{i=1}^n y_i^{(KM)+} \alpha_i^{(KM)+} \Phi(\mathbf{x}_i) \quad (3.123)$$

where the coefficients $\{\alpha_i^{(KM)+}\}$ are associated with cluster-related provisional classes, $y_i^{(KM)+}$.

Then one builds up the dual training set, which supports the opposite labeling schema:

$Z^- = \{(\mathbf{x}_i, y_i^{(KM)-}); i = 1, \dots, n; y_i^{(KM)-} = 1 - 2m_i\}$, and obtains the alternative parameters, $\mathbf{w}^{(KM)-}$, as:

$$\mathbf{w}^{(KM)-} = -\mathbf{w}^{(KM)+} \quad (3.124)$$

To choose between the two alternatives $\{\mathbf{w}^{(KM)+}, \mathbf{w}^{(KM)-}\}$, one follows the Structural Risk Minimization principle, and picks the labeling schema that better constrains complexity. If one denotes with $\mathbf{w}^{(TG)}$ the solution obtained by using real labels, and with $\mathbf{w}^{(KM)}$ the unsupervised “reference” solution, the latter parameter set that further constrains SVM capacity is:

$$\mathbf{w}^{(KM)} = \arg \min(\|\mathbf{w}^{(KM)+} - \mathbf{w}^{(TG)}\|, \|\mathbf{w}^{(KM)-} - \mathbf{w}^{(TG)}\|) \quad (3.125)$$

Upon completion of the unsupervised analysis, it is convenient to use the reference solution, $\mathbf{w}^{(KM)}$, and the artificial target settings, $y^{(KM)}$, adopted to attain the unsupervised SVM training, to compute the following quantities:

$$\beta_i = y_i \sum_{j=1}^n \alpha_j^{(KM)} y_j^{(KM)} K(\mathbf{x}_i, \mathbf{x}_j) \quad i = 1, \dots, n \quad (3.126)$$

This set of parameters will be used later in the theoretical treatment.

3.4.1.2 Using unsupervised clustering to constrain SVM capacity during bound computation

The result (3.125) of the unsupervised analysis is used in the computation of the MD bound on random targets, and sets a reference to arrange the family of classifiers within the hypothesis space, \mathcal{F} .

The quantity ruling the ordering of classifiers is the distance, in the weight

space, between a given SVM solution, \mathbf{w} , and the reference configuration, $\mathbf{w}^{(KM)}$:

$$\rho_w = \|\mathbf{w} - \mathbf{w}^{(KM)}\| \quad (3.127)$$

Such an unsupervised-reference approach offers a straightforward interpretation: whenever the result of clustering matches the true distribution of pattern classes, the unsupervised separation surface, $\mathbf{w}^{(KM)}$, and the real classification surface, $\mathbf{w}^{(TG)}$, must coincide. Of course, the opposite case may occur, in which the target distribution is totally uncorrelated with the obtained clusters; The varying displacements obtained from empirical results can give useful information about the complexity of the specific classification problem. Such a distance-based ordering is profitable in any bound computation method. For a given value of ρ_w , only the classifier configurations lying within the hypersphere having radius ρ_w , and centered in $\mathbf{w}^{(KM)}$, will be considered to compute the complexity term of the MD generalization bound. Larger and larger spheres enable the training algorithm to pick the optimal weight set, \mathbf{w} , from among wider and wider families of classifiers. As the chance of fitting the various random target settings increases, the associated generalization bounds widen accordingly. The radius, ρ_w , of the sphere is the crucial quantity driving the SRM principle, and the proper setting of such a regularization parameter is of paramount importance. The VQSVM approach, in order to be an auxiliary machine for SVM, uses the SVM solution, $\mathbf{w}^{(TG)}$, obtained on the real labels to delimit the valid portion of the weight space for the admissible solutions. The regularization parameter is locked and set to:

$$\rho_0 \equiv \|\mathbf{w}^{(TG)} - \mathbf{w}^{(KM)}\| \quad (3.128)$$

In other words, every solution, \mathbf{w} , obtained during the MD estimation process must obey the distance-based criterion:

$$\rho_w \leq \rho_0 \quad (3.129)$$

and the optimization problem to be solved is expressed by the usual SVM cost under the additional Ivanov-like constraint (3.129).

The choice of this regularization parameter is critical: choosing ρ_0 in this way means that whenever VQSVM is trained on original labels y , then the original solution $\mathbf{w}^{(TG)}$ can be re-obtained: thus VQSVM for original labels is equivalent to SVM because the constraint is not active; the constraint activates whenever other labels, such as random labels of MD, are used; this allows a different behaviour of SVM and VQSVM when computing the bound. The more the space of solutions is constrained the less is the VQSVM fitting capability. One notes that ρ_0 should be set a priori, instead here ρ_0 is a function of both X and y ; strictly speaking observing y invalidates the theory supporting generalization error bounds; to be fully theoretical compliant the samples used to set \mathbf{w}^{KM} should be different from that used for learning; despite this the aim here is assessing the shrinking of the bounds and final empirical effectiveness of the approach is the goal; moreover using one single set of data is more practical. Roughly speaking one uses an aggressive, not fully theoretically justified, strategy to shrink bounds and obtain effective model selection.

This method, moreover, poses two major questions. The first question regards the effectiveness of constraint (3.129) in bounding the generalization error. In the most favorable case, one has $\mathbf{w}^{(TG)} \equiv \mathbf{w}^{(KM)}$: the separating surface drawn by unsupervised clustering on artificial targets coincides with that obtained with real targets, hence one has: $\max\{\rho_w\} = 0$. When the empirical classifier matches the natural distribution of data, the number of allowed family members reduces to one, and the associate complexity term, $(C_{\mathcal{F}})$, theoretically vanishes. By contrast, the worst situation occurs when $\|\mathbf{w}^{(TG)} - \mathbf{w}^{(KM)}\| \approx \infty$. As the hypersphere encompasses the entire weight space, all classifiers in \mathcal{F} are admissible, hence one must pay the price of testing the whole set of alternatives within the family. In this case, the generalization bound gets back to the basic prediction provided by classical Structural Risk Minimization. This proves that the VQSVM approach is consistent with (and, on average, improves on) the conventional sample-based SRM, whose prediction is taken as the worst-case option.

The second, operational question concerns the availability of an effective op-

timization process to solve the reformulated problem SVM+(3.129). The crucial issue is that (3.129) involves a quadratic constraint, hence the optimization problem cannot be expressed any longer as a conventional SVM training. The following sections derive an iterative approach to the augmented formulation SVM+(3.129), which can still take advantage of a Quadratic-Programming formulation and ensure convergence to the global, optimal solution.

3.4.1.3 Including the additional constraint into the SVM optimization problem

The modified Primal formulation includes the quadratic constraint (3.129) in the SVM basic problem setting, and can be written as:

$$\min_{\mathbf{w}, b, \xi} P_M = \frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \{y_i(\mathbf{w}\mathbf{x}_i + b) - 1 + \xi_i\} - \sum_{i=1}^n \gamma_i \xi_i - \frac{\lambda}{2} [\rho_0^2 - \|\mathbf{w} - \mathbf{w}^{(KM)}\|^2] \quad (3.130)$$

where P_M stands for Modified Primal, and λ is the Lagrange multiplier associated to the constraint (3.129). One derives a Dual problem formulation by nullifying the partial derivatives with respect to the optimization variables (x^i indicates the i -th component):

$$\begin{aligned} \frac{\partial P_M}{\partial w^i} &= w^i - \sum_{j=1}^n \alpha_j y_j \Phi(x_j)^i + \lambda(w^i - w^{(KM)i}) = 0 \\ \frac{\partial P_M}{\partial b} &= \sum_{j=1}^n \alpha_j y_j = 0 \\ \frac{\partial P_M}{\partial \xi_i} &= C - \alpha_i - \gamma_i = 0 \end{aligned} \quad (3.131)$$

By solving the first KKT on \mathbf{w} :

$$\mathbf{w} = \frac{\sum_{i=1}^n \alpha_i y_i \Phi(\mathbf{x}_i) + \lambda \mathbf{w}^{(KM)}}{(1 + \lambda)} \quad (3.132)$$

When the multiplier λ is zero, constraint (3.129) is inactive and the solution, \mathbf{w} , takes back the form of the basic SVM parameters; this means that the solution lies inside the sphere (3.129). In the following, the symbol $\mathbf{w}^{(\lambda=0)}$ will denote the weight vector associated with this case:

$$\mathbf{w}^{(\lambda=0)} = \sum_{i=1}^n \alpha_i y_i \Phi(\mathbf{x}_i) \quad (3.133)$$

Expression (3.133) holds for any class configuration $\{y_i\}$, and embeds the part of the solution which does not consider the quadratic constraint (3.129). Rewriting problem P_M into its Dual formulation leads to the optimization problem:

$$\min_{\alpha, \lambda} D_M = \left(\frac{\|\mathbf{w}^{(\lambda=0)}\|^2}{2} - \sum_i \alpha_i \right) + \frac{\lambda}{2} \left[\rho_0^2 - \frac{\|\mathbf{w}^{(\lambda=0)} - \mathbf{w}^{(KM)}\|^2}{(1+\lambda)} \right] \stackrel{def}{=} D_{M, SVM} + D_{M, \lambda}$$

$$\text{subject to : } \begin{cases} 0 \leq \alpha_i \leq C & \forall i = 1, \dots, n \\ \sum_i \alpha_i y_i = 0 & \forall i = 1, \dots, n \\ \lambda \geq 0 \end{cases}$$

(3.134)

The cost function in (3.134) comprises two terms: the left term, denoted as $D_{M, SVM}$, identifies the portion of the total cost that only depends on parameters α_i ; it coincides with the ‘classical’ SVM Dual cost. The additional right term, defined as $D_{M, \lambda}$, is parameterized by λ and takes into account the contribution of the quadratic constraint:

$$\min_{\alpha, \lambda} D_M = \min_{\alpha, \lambda} [D_{M, SVM}(\alpha) + D_{M, \lambda}(\alpha, \lambda)]$$

(3.135)

After simple derivations and substitutions the dual optimization problem is eventually written as:

$$\min_{\alpha, \lambda} \left\{ \left(\frac{\|\mathbf{w}^{(\lambda=0)}\|^2}{2} - \sum_{i=1}^n \alpha_i \right) + \frac{\lambda}{2} \left[\rho_0^2 - \frac{\|\mathbf{w}^{(\lambda=0)} - \mathbf{w}^{(KM)}\|^2}{(1+\lambda)} \right] - \sum_i \mu_i \alpha_i - \sum_i \xi_i (C - \alpha_i) \right. \\ \left. + b \sum_i \alpha_i y_i - \delta \lambda \right\}$$

(3.136)

The additional variables in (3.136) embed the constraints (3.129); δ ensures non-negative values of the basic Lagrange multiplier, λ . To find the solution to the Modified Dual, one first computes the (classical) SVM solution without the additional constraint, then check if condition (3.129) is fulfilled; this may result in three different cases, depending on the position of the solution vector, \mathbf{w} .

- **Case 1)** (3.17 first) The solution lies inside the sphere (3.129), hence the

constraint is inactive and $\lambda = 0$. The Modified problem (3.134) reduces to the conventional form, and the vector \mathbf{w} is a valid solution.

- **Case 2)** (3.17 second) Both λ and δ vanish, hence the SVM solution lies *exactly* on the sphere surface and the constraint is inactive. In this case, too, the solution \mathbf{w} is a valid solution.
- **Case 3)** (3.17 third) The solution is out of the sphere, $\lambda > 0$, the constraint is active, and one has:

$$\|\mathbf{w} - \mathbf{w}^{(KM)}\|^2 > \rho_0^2 \quad (3.137)$$

In this case, to reach a valid solution one requires an ad-hoc optimization process, which will be presented in the following Section.

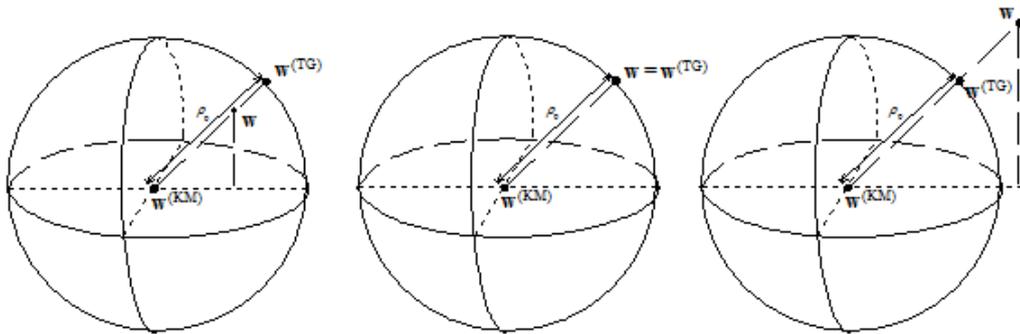


Figure 3.17: Relative positions of the solution vector, \mathbf{w} , with respect to the unsupervised reference, $\mathbf{w}^{(KM)}$, from left: Case 1): Within the hypersphere; Case 2): on the hypersurface; Case 3): Out of the hypersphere

3.4.2 Algorithm for Constrained Optimization

The optimization of the training problem involved by eq.(3.134) is in fact straightforward. As it will be shown in the following, the problem is convex and allows one to reach a global solution, since any gradient descent-based algorithm can support the optimization task. This Section proposes a simple two-step procedure that optimizes (3.134) and relies on any off-the-shelf

SVM optimizer (e.g. SMO [60]), thus yielding a straightforward algorithmic implementation.

3.4.2.1 Optimization theory for the modified Dual problem

A prerequisite to ensure that the minimum of D_M can always be found is to verify that the functional (3.134) is convex. The following Theorem confirms this fact by proving that the associate Hessian matrix, \mathbf{H} , is positive semi-definite. It is convenient in the following to use a compact notation and define the matrix \mathbf{Q} , having size $n \times n$, as the matrix composed by the elements:

$$q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (3.138)$$

Theorem 3.4.1. *The Modified Dual functional (3.134) is convex, and its minimization implies a convex optimization problem that admits a global solution.*

Proof: for the sake of brevity, the proof is given in the Appendix. The Modified Dual cost does not benefit from the straightforward quadratic form that characterizes conventional SVMs, hence no classical SVM-training algorithm applies directly to solve (3.134) under Case 3). The VQSVM framework includes an ad-hoc algorithm that offers two advantages: it ensures convergence to the global minimum, and it allows one to reuse efficient SVM training algorithms. The algorithm proceeds by alternating two steps. The first step minimizes $D_{M,\lambda}$ and works out the optimal value, $\tilde{\lambda}$, of the Lagrange multiplier when the parameters, α , remain fixed in (3.134). The following Lemma gives the analytical expression of $\tilde{\lambda}$:

Lemma 3.4.1. *The optimal value, $\tilde{\lambda}$, that minimizes $D_{M,\lambda}$ for a fixed set of parameters, α , is:*

$$\tilde{\lambda} = \sqrt{\frac{\alpha^t \mathbf{Q} \alpha + \|\mathbf{w}^{(KM)}\|^2 - 2[\mathbf{w}^{(\lambda=0)}]^t \mathbf{w}^{(KM)}}{\rho_0^2}} - 1. \quad (3.139)$$

Proof: the proof is given in the Appendix.

The second step minimizes the functional (3.134) over the parameters, α , while $\tilde{\lambda}$ keeps fixed. The following Lemma proves that, in the latter case, the cost takes on the typical formulation of SVM with a minor correction to the linear term; this allows one to use efficient, conventional algorithms for SVM training to support the second step.

Lemma 3.4.2. *For a fixed value $\tilde{\lambda}$, the quadratic convex cost D_M to be minimized can be written as:*

$$\begin{aligned} & \frac{1}{2} \alpha^t \mathbf{Q} \alpha - \sum_{i=1}^n (1 + \tilde{\lambda}(1 - \beta_i)) \\ & 0 \leq \alpha_i \leq C \quad \forall i = 1, \dots, n \\ & \sum_{i=1}^n \alpha_i y_i = 0 \quad \forall i = 1, \dots, n \end{aligned} \tag{3.140}$$

Proof: the proof is given in the Appendix. In the following, \hat{b} will denote the Lagrange multiplier associated with the linear constraint in problem (3.134); this multiplier is worked out as a side result of the minimization process of (3.134). The procedure alternating Lemma 3.4.1 and Lemma 3.4.2 iterates until a solution lying within the hypersphere (3.129) is found. The following Theorem proves that such a procedure always converges to the global minimum.

Theorem 3.4.2. *A procedure alternating the partial optimizations as per Lemma 3.4.1 and Lemma 3.4.2, always reaches the global minimum of the Modified Dual cost eq.(3.134).*

Proof Lemma 3.4.1 and Lemma 3.4.2 ensure the minimization of $D_{M,SVM}$ and $D_{M,\lambda}$ when the multiplier $\tilde{\lambda}$ and the parameters, α , remain constant, respectively. This implies that at least one term in the summation (3.134) decreases. Therefore, at the i -th iteration alternating Lemma 3.4.1 and Lemma 3.4.2, one always has: $D_M^{(i)} < D_M^{(i-1)}$, and the process necessarily minimizes cost (3.134). Theorem proves that that cost is convex, hence the minimum is global, and the sequence $\{D_M^{(1)}, D_M^{(2)}, D_M^{(3)}, \dots, D_M^{(n)}\}$ converges to the global minimum of (3.134). The following pseudocode summarizes the overall optimization procedure:

In the pseudocode, τ is a tolerance threshold to detect when the solution is close enough to the surface of the hypersphere.

Algorithm 4 VQ-SVM Optimization Algorithm for the Modified Dual**Require:** Hessian matrix \mathbf{Q} , ρ_0 , C , y , τ **Ensure:** Vector α , λ

- 1: $\alpha = \arg \min D_{M,SVM}, \lambda = 0$
- 2: **while** $|\|\mathbf{w} - \mathbf{w}^{(KM)}\|^2 - \rho_0^2| \leq \tau$ **do**
- 3: $\tilde{\alpha} = \arg \min_{\alpha} D_M(\tilde{\lambda})$ (3.140)
- 4: $\tilde{\lambda} = \arg \min_{\lambda} D_{M,\lambda}(\tilde{\alpha})$ (3.139)
- 5: **end while**

3.4.2.2 Operational aspects in the optimization procedure

The most computation-intensive phases of the above algorithm are the optimization processes at. It has been proved [101] that the computational cost of SMO is roughly a quadratic function of the number of patterns; the computational cost of (3.139) is a quadratic function of the number of support vectors that result from the modified SVM solution. Thus, if one denotes with k the number of iterations of the optimization algorithm, the complexity can be worked out as:

$$O((n_{sv})^2 + kn^2) \quad (3.141)$$

Since the iterations are performed on the same dataset, the kernel matrix can be computed once and offline. The tolerance-based stopping criterion is not critical, as with a typical tolerance threshold $\tau = 1e - 3$ the quadratic constraint is found to be satisfied easily. The only crucial aspect in the procedure is to attain a ‘precise’ solution in the SVM-based inner loop; this means that the Karesh-Kuhn-Tucker conditions must be fulfilled with a tolerance no larger than τ .

Finally, an important aspect concerns the decision function to classify new input patterns. This function can be obtained from problem D_M by using (3.132), and is written as:

$$\begin{aligned} & \text{sign}(\mathbf{w} \cdot \Phi(\mathbf{x}_j) + b) = \\ & = \text{sign} \left(\frac{\sum_i \alpha_i y_i \Phi(x_i) + \lambda \mathbf{w}^{(KM)}}{(1+\lambda)} \cdot \Phi(\mathbf{x}_j) + b \right) \end{aligned} \quad (3.142)$$

or equivalently:

$$\text{sign}(\mathbf{w} \cdot \Phi(\mathbf{x}_j) + b) = \text{sign}\left(\frac{1}{1+\lambda} \left(\sum_{i=1}^n \alpha_i y_i K_{ij} + \lambda \sum_{i=1}^n \alpha_i^{(KM)} y_i^{(KM)} K_{ij}\right) + b\right) \quad (3.143)$$

When $\lambda = 0$, the regularization is no more biased and the problem reduces to the classical SVM decision function. The bias term in eq.(3.143) obeys the following Lemma:

Lemma 3.4.3. *Let \hat{b} be the Lagrange multiplier derived from minimizing problem (3.134): then the bias b appearing in the decision function (3.143) is $b = \hat{b}/(1 + \lambda)$.*

Proof: for clarity, the proof is given in the Appendix.

3.4.2.3 A synthetic review

The basic idea to control the complexity of a Support Vector Machine is to reduce the space of admissible classifiers by a (sample-based) reference solution. In the present approach, constraint (3.129) implements an unsupervised-based reference criterion. The resulting, additional constraint turns the conventional SVM learning process into a Quadratic-Constrained, Quadratic-Programming optimization problem. The theoretical and practical frameworks prove that the described approach solve that problem effectively, as it can find the global minimum and, at the same time, re-use existing efficient algorithms for SVM training. One can efficiently use VQSVM (instead of SVM) within the computation of the Maximal-Discrepancy bound, and attain model-selection results that always are equal (at worst) or better than those provided by classical Structural Risk Minimization.

3.4.3 Experimental Results

The empirical validation of the proposed method involved extensive experiments on real-world datasets, namely, the MNIST numerical recognition testbed [99], text-classification problems drawn from the “Newsgroup-20” dataset [100]; reference UCI datasets “Heart”, “Ionosphere”, “Sonar”, and “Pima Indian Diabetes” [100] were also tested.

3.4.3.1 Experimental procedure

A version of SMO with RBF kernel supported the model presented in [62] with first-order selection of the working set. In the iterated procedure, the tolerance value for both SMO training and the Vector-Quantization constraint-based version (see pseudocode) always was $\tau = 1e - 3$. The model-selection approach scanned wide ranges of hyper-parameter settings; this led to a huge number of SVM training cycles. For each cycle, the input quantities were:

- the specific settings of the SVM hyper-parameters $\{C, \sigma\}$;
- a training set, Z , of labeled data. In the bound estimation, classes were set at random in compliance with the MD procedure.
- a validation set, Z_{VAL} , of patterns whose labels always coincided with the true classes, which was used to verify the accuracy in predicting generalization performance.

The latter steps allowed one to compare directly the two bound estimates, and to get a sound verification of the effectiveness of the additional constraint in two ways: first, by assessing the contribution of the VQ-induced constraint in shrinking the theoretical bound; secondly, by measuring the actual reduction in classification error on unseen data.

The outputs of the procedure included:

- the generalization error bound, $R^{(TG)}$, that resulted from summing the training SVM error, R_{emp} , with the conventional MD-based penalty term computed as usual;
- the generalization error bound, $R^{(VQ-TG)}$, that resulted from summing the training SVM error, R_{emp} , with the MD-based penalty term subject to the quadratic constraint (3.129);
- the “true” empirical generalization error, R_{VAL} , measured on the validation set, Z_{VAL} .

Algorithm 5 VQSVM Experimental Protocol

Require: Training set Z , validation set Z_{VAL} hyperparameters, C , σ^2 , n , $NumIter$

Ensure: Generalization error estimations: $R^{(TG)}, R^{(VQ-TG)}, R_{VAL}$.

- 1: Build the kernel matrix \mathbf{K} on training data, Z .
- 2: Train an SVM on training data (original classes) $\rightarrow \{\mathbf{w}^{(TG)}, b\}$
- 3: Compute the empirical training error $R_{emp}^{(TG)}$, using \mathbf{y} , \mathbf{K} $\{\mathbf{w}^{(TG)}, b\}$
- 4: Perform clustering of data in Z , $\rightarrow \{\mathbf{m}, \Psi_0, \Psi_1\}$
- 5: Apply an artificial labeling schema: $y^{(KM)+} \equiv \{\Psi_0 \rightarrow +1, \Psi_1 \rightarrow -1\}$
- 6: Train an SVM on training data and artificial labels $y^{(KM)+} \rightarrow \{\mathbf{w}^{(KM)\pm}, b^{(KM)\pm}\}$.
- 7: Select unsupervised reference $\mathbf{w}^{(KM)}, y^{(KM)}, b^{(KM)}$ according to (3.125)
- 8: Compute Maximal Discrepancy generalization bounds:
- 9: $\hat{R}^{MD} = 0, \hat{R}^{VQ-MD} = 0$.
- 10: **for** $i = 1 : NumIter$ **do**
- 11: Apply random-swap labeling schema $y^{(MD)}$
- 12: Train an SVM on training data using classes $y^{(MD)} \rightarrow \mathbf{w}^{(MD)}, b^{(MD)}$
- 13: Compute MD training error \hat{R}_{emp} using $\mathbf{w}^{(MD)}, b^{(MD)}$ and $y^{(MD)}$.
- 14: Train a constrained SVM on Z with classes $y^{(MD)} \rightarrow \mathbf{w}^{(VQ-MD)}, b^{(VQ-MD)}$
- 15: Compute the MD training error $\hat{R}_{emp}^{(VQ)}$ using $\mathbf{w}^{(VQ-MD)}, b^{(VQ-MD)}$ and $y^{(MD)}$
- 16: $\hat{R}^{(MD)} = \hat{R}^{(MD)} + \hat{R}_{emp}$
- 17: $\hat{R}^{(VQ-MD)} = \hat{R}^{(VQ-MD)} + \hat{R}_{emp}^{(VQ)}$
- 18: **end for**
- 19: $\hat{R}^{(MD)} = \hat{R}^{(MD)} / NumIter$
- 20: $\hat{R}^{(VQ-MD)} = \hat{R}^{(VQ-MD)} / NumIter$
- 21: Compute generalization error bound (conventional SVM):
- 22: $R^{(TG)} = \hat{R}_{emp}^{(TG)} + (1 - 2\hat{R}^{(MD)})$
- 23: Compute generalization error bound (constrained SVM):
- 24: $R^{(VQ-TG)} = \hat{R}_{emp}^{(TG)} + (1 - 2\hat{R}^{(VQ-MD)})$
- 25: Evaluate validation error (conventional SVM) R_{VAL} on Z_{VAL} using $\mathbf{w}^{(TG)}, b$.

3.4.3.2 MNIST Experiments

The MNIST testbed involved a 10-digit character recognition problem. The experimental procedure covered all pairs of digits exhaustively, thus involving 45 independent problems. For each problem, the training set included 200 patterns; at the same time, each experiment included a validation set holding a separate group of 6000 patterns. Using a small training set and a much larger test set made it possible to verify the method's effectiveness in a limited-sample scenario, yet benefiting from a reliable estimate of the true generalization error.

To complete model selection for each binary problem, the training process was repeated for a set of hyper-parameter settings, $\{C, \sigma\}$, whose admissible values were: $C \in \{1, 10, 10^2, 10^3, 10^4\}$, and $2\sigma^2 \in \{10^2, 10^3, 10^4, 10^5, 10^6\}$. This required to solve about one million of QP problems, each having complexity $n = 200$. Each row in Table 3.2 addresses one of the 45 binary OCR problems and gives:

1. the most promising hyper-parameters, $(C, 2\sigma^2)$, resulting from the model-selection process;
2. the associate generalization bounds, for both the conventional and the constrained MD approach;
3. the validation error, $R_{VAL}^{(TG)}$, measured when using the model selection suggested by the conventional MD bound;
4. the validation error, $R_{VAL}^{(VQ-TG)}$, measured when using the model selection suggested by the constrained-SVM MD bound.

The constrained approach almost always yielded tighter generalization bounds. Even more importantly, the validation errors reduced accordingly, thus showing that the unsupervised reference also led to a better model selection. To demonstrate the different properties of either approach, Figure 3.18 gives the progression of the generalization bound (in the hyper-parameter space) for the problem '1' vs '7', which proved quite difficult due to the similar appearances of the digits. The graphs clearly show that the constrained ap-

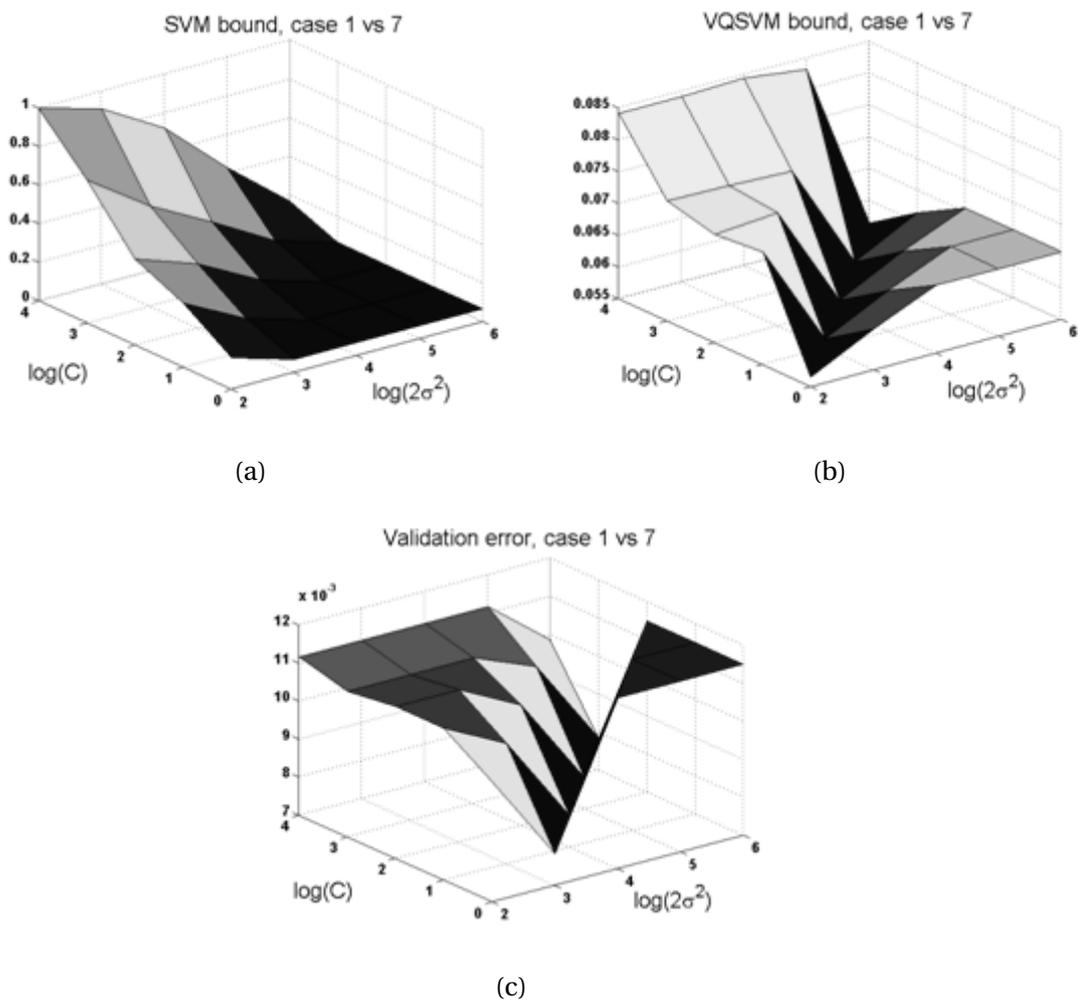


Figure 3.18: Model selection surfaces in the hyper-parameter space: a) Conventional-SVM MD-bound surface . b) Constrained-SVM MD bound surface c) Validation-error surface

proach succeeded in approximating accurately the *shape* of the validation-error surface; moreover, the predicted bounds always were lower than their conventional-SVM counterparts. Similar results were obtained for the other digit-pair problems.

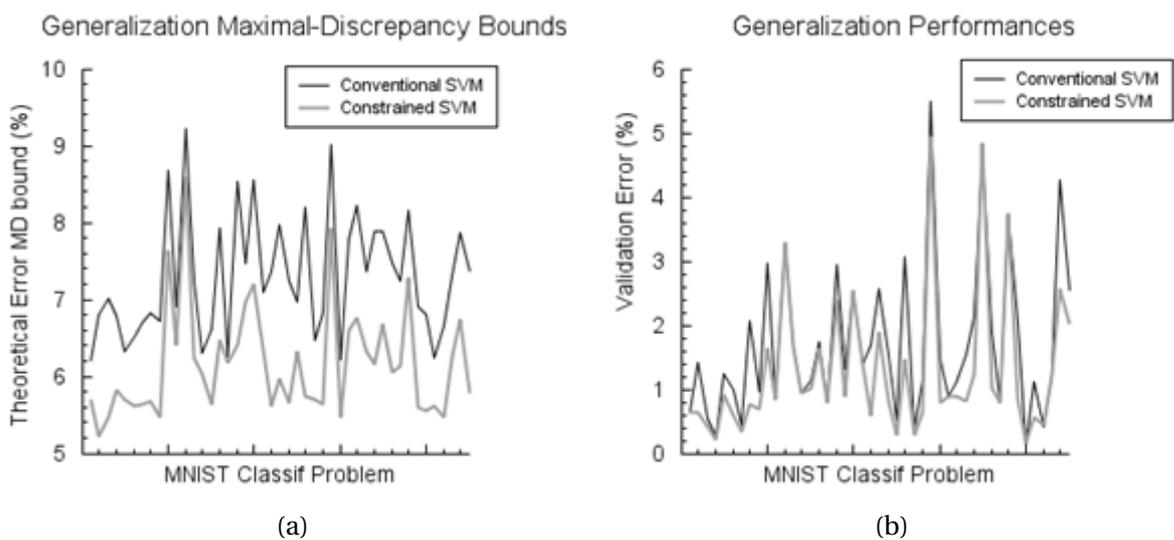


Figure 3.19: Comparison of conventional and constrained SVM model selection methods. a) Generalization bounds b) Validation error

The effectiveness of the constrained-SVM method can also be assessed by analyzing the results of Table 3.2 in a graphical way. Figure 3.19 compares the theoretical generalization bounds, one derived with a conventional Maximal-Discrepancy procedure, the other obtained by including the additional constraint. The graph shows that the latter, constraint-based prediction always kept lower than the conventional bound. A similar result was obtained when measuring the run-time error performances on unseen validation data; the curves for both approaches (3.19b) confirmed the effectiveness of the constraint-based approach, as the constrained classifier model selection method further reduced the actual generalization error.

An intriguing aspect of the obtained results was that the hyper-parameters selected by VQSVM seemed more 'aggressive' than those selected by conventional MD bounds on SVM. Maximum Discrepancy tends to select hyper-parameters that are prone to under-fitting data; by using VQSVM such a trend

was mitigated. As compared with the conventional model-selection procedure, the hyper-parameters settings prompted by VQSVM were larger in C and/or smaller in σ . As a consequence, the deviations in both hyper parameters contributed to make up for under-fitting.

The VQSVM approach cannot yield any analytical prediction of the eventual hyper-parameter settings, yet the overall model behavior seems to suggest some global, marked trend that might be useful in a classifier design process. In particular, VQSVM seems to provide a good countermeasure against data under-fitting; from a general viewpoint, the model-selection results resembled those obtained when using large samples of patterns and/or applying k-fold cross-validation to make robust estimates.

3.4.3.3 Newsgroup-20 experiments

This experimental campaign involved bi-class problems from the Newsgroup-20 dataset for text mining [100]. Patterns were represented by text documents, whose pre-processing phases included stop-words removal (i.e., the elimination of semantically non-selective expressions from the text) and Porter's word stemming [102] (to group derived lexical instances). Thus, a document D eventually consisted in a sequence of significant tokens called 'index terms'. The associate vector-space model [103] spanned a T -dimensional dictionary, and represented each document D as a vector of real-valued weight terms. Each component of the T -dimensional vector is a non-negative weight term that denotes the relevance of the term itself within the document D , (e.g., its term frequency). The text processing adopted the methods and paradigms presented in [17]. The experiments involved the following five binary classification problems: sci.electronics VS rec.sport.baseball, sci.space VS sci.med, alt.atheism VS sci.crypt, rec.sport.hockey VS rec.sport.baseball, talk.politics.guns VS talk.politics.mideast. Table 3.3 summarizes the experimental set-up for the involved tests. For each problem, a set Z of 200 training documents were randomly drawn for each class; the remaining documents were used to measure the generalization error. The settings of hyper-parameters were $C \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 102, 103\}$, and $2\sigma^2 \in \{1, 10, 102, 103, 104, 105\}$.

The reported results confirmed some significant properties that have been

<i>Problem</i>	Conventional SVM		Constrained SVM		Theoretical Bound		Validation Error	
	$C^{(TG)}$	$2[\sigma^{(TG)}]^2$	$C^{(VQ-TG)}$	$2[\sigma^{(VQ-TG)}]^2$	$R^{(TG)}$	$R^{(VQ-TG)}$	$R_{VAL}^{(TG)}$	$R_{VAL}^{(VQ-TG)}$
0 vs 1	1	1e6	1e3	1e6	6.20%	5.70%	0.65%	0.63%
0 vs 2	1	1e6	1	1e2	6.80%	5.23%	1.42%	0.65%
0 vs 3	1	1e6	1e4	1e6	7.01%	5.47%	0.55%	0.42%
0 vs 4	1	1e6	1	1e2	6.78%	5.83%	0.27%	0.22%
0 vs 5	1	1e6	1e3	1e6	6.32%	5.70%	1.23%	0.92%
0 vs 6	1e3	1e6	1e2	1e4	6.51%	5.62%	0.98%	0.60%
0 vs 7	1	1e6	1	1e2	6.69%	5.64%	0.43%	0.35%
0 vs 8	1	1e6	10	1e3	6.83%	5.68%	2.07%	0.77%
0 vs 9	1	1e6	1e4	1e6	6.73%	5.48%	0.97%	0.68%
1 vs 2	1	1e6	1e3	1e6	8.68%	7.64%	2.97%	1.65%
1 vs 3	1	1e4	1e3	1e6	6.91%	6.40%	0.87%	0.83%
1 vs 4	10	1e4	1	1e3	9.21%	8.59%	3.28%	3.28%
1 vs 5	1	1e3	1	1e3	7.26%	6.24%	1.67%	1.67%
1 vs 6	1	1e4	1	1e6	6.30%	6.04%	0.95%	0.95%
1 vs 7	1	1e6	1e4	1e6	6.62%	5.64%	1.15%	1.02%
1 vs 8	10	1e4	1	1e2	7.92%	6.48%	1.73%	1.65%
1 vs 9	1	1e3	1	1e3	6.27%	6.17%	0.80%	0.80%
2 vs 3	1	1e3	1e4	1e6	8.54%	6.40%	2.93%	2.38%
2 vs 4	1	1e6	1e3	1e6	7.47%	6.96%	1.32%	0.90%
2 vs 5	10	1e4	1e3	1e6	8.55%	7.19%	2.53%	2.53%
2 vs 6	1	1e3	1e3	1e6	7.10%	6.34%	1.40%	1.40%
2 vs 7	1	1e6	10	1e3	7.36%	5.61%	1.68%	0.58%
2 vs 8	1e3	1e6	1	1e2	7.97%	5.98%	2.57%	1.90%
2 vs 9	1	1e6	1	1e2	7.25%	5.65%	1.55%	0.82%
3 vs 4	1	1e6	1e3	1e6	6.97%	6.33%	0.52%	0.30%
3 vs 5	1	1e3	1	1e2	8.19%	5.74%	3.07%	1.47%
3 vs 6	1	1e6	1e3	1e6	6.46%	5.70%	0.38%	0.28%
3 vs 7	1	1e6	10	1e3	6.82%	5.64%	1.15%	0.63%
3 vs 8	1	1e6	1e3	1e6	9.02%	7.92%	5.50%	4.95%
3 vs 9	1	1e6	1e4	1e6	6.22%	5.48%	1.47%	0.80%
4 vs 5	1e2	1e5	1e3	1e6	7.79%	6.59%	0.88%	0.88%
4 vs 6	10	1e4	1	1e2	8.23%	6.76%	1.12%	0.88%
4 vs 7	1	1e6	1e3	1e5	7.36%	6.32%	1.55%	0.82%
4 vs 8	1	1e6	10	1e3	7.88%	6.16%	2.15%	1.25%
4 vs 9	1e2	1e5	1	1e3	7.89%	6.67%	4.77%	4.83%
5 vs 6	1	1e6	1e4	1e6	7.47%	6.05%	1.90%	1.02%
5 vs 7	10	1e4	1e3	1e6	7.24%	6.14%	0.78%	0.78%
5 vs 8	1	1e3	1	1e3	8.16%	7.28%	3.73%	3.73%
5 vs 9	1	1e6	1e3	1e2	6.91%	5.59%	2.05%	0.83%
6 vs 7	1	1e6	10	1e3	6.80%	5.56%	0.13%	0.17%
6 vs 8	1	1e6	1e4	1e6	6.25%	5.62%	1.12%	0.57%
6 vs 9	1	1e6	1	1e2	6.65%	5.48%	0.42%	0.43%
7 vs 8	10	1e4	1e3	1e6	7.31%	6.25%	1.22%	1.22%
7 vs 9	1	1e6	1	1e2	7.86%	6.75%	4.27%	2.57%
8 vs 9	1	1e3	1e4	1e6	7.36%	5.79%	2.55%	2.02%

Table 3.2: MNist digit recognition. Model-Selection results, bounds, and accuracy of conventional svm and constrained svm.

<i>Problem</i>	<i>Dataset</i>	<i>#Training</i>	<i>#Test</i>
1	<i>sci.electronics VS rec.sport.baseball</i>	200	1775
2	<i>sci.space VS sci.med</i>	200	1777
3	<i>alt.atheism VS sci.crypt</i>	200	1590
4	<i>rec.sport.hockey VS rec.sport.baseball</i>	200	1793
5	<i>talk.politics.guns VS talk.politics.mideast</i>	200	1650

Table 3.3: Newsgroup-20 binary problems

<i>Problem</i>	<i>Conventional SVM</i>		<i>Constrained SVM</i>		<i>Theoretical Bound</i>		<i>Validation Error</i>	
	$C^{(TG)}$	$2[\sigma^{(TG)}]^2$	$C^{(VQ-TG)}$	$2[\sigma^{(VQ-TG)}]^2$	$R^{(TG)}$	$R^{(VQ-TG)}$	$R_{VAL}^{(TG)}$	$R_{VAL}^{(VQ-TG)}$
1	1	1	1000	10	24.0%	6.3%	24.0%	5.9%
2	1	1	1000	10	8.6%	6.4%	8.1%	4.9%
3	10	1000	10	1	16.2%	5.4%	12.5%	4.1%
4	10	1	1000	10	34.0%	7.8%	7.4%	7.2%
5	10	1	1000	10	16.89%	5.3%	4.24%	3.64%

Table 3.4: Newsgroup-20 results

observed in the MNIST experiments. First, the conventional and VQSVM-based generalization estimates lead to different model selection outcomes, and, overall, the latter method leads to higher settings of hyper-parameter C . This witnessed the fact that the models chosen by the VQSVM model appear less conservative as compared with the classical one. More importantly, for the Newsgroup-20 testbed, too, a significant reduction in the bound values coincided with more effective model-selection choices in matching the actual generalization errors.

3.4.3.4 Experiments on limited-sample UCI datasets

The experimental verification of the VQSVM approach involved an additional set of reference datasets from UCI [100], namely, Spec “Heart”, “Sonar”, Pima Indian “Diabetes”, and “Ionosphere”. In all cases the testbeds were chosen mainly for their particular (limited) distributions of the training sets, in the presence of possibly intricate decision surfaces.

The experiments involving Spec “Heart” maintained the original partitions of data, including 80 training patterns and 187 test patterns. The “Sonar” sample was used for training entirely, due to the very small number of patterns ($n = 208$). The patterns for Pima Indian “Diabetes” were randomly split into a training set holding 230 patterns, and a test set including 538 patterns. The “Ionosphere” dataset was split into a training set of 251 patterns and a test set of 100 test patterns.

In all testbeds, the model-selection grid of tested hyper-parameter settings was made by: $C \in \{0.01, 0.1, 1, 10, 100\}$ $2\sigma^2 \in \{10^{-2}, 10^{-1}, 1, 10\}$. Tables 3.5,3.6,3.7,3.8, 3.9 presents the obtained results; The Tables compare the predictions for both classical and VQSVM-based error bounds, and report, whenever possible, the outcomes of the model-selection processes in terms of measured generalization performance.

Empirical evidence highlights the complex nature of the classification problems involved; in the VQSVM framework that complexity partially invalidated the “cluster assumption”. This feature, in conjunction with the limited empirical sample, led to bound values that were objectively quite high for both the classical MD and the VQSVM-based approach.

As expected from theory, even in such this situation (where the fundamental cluster hypothesis is not completely or partially true) the bounds predicted by the VQSVM paradigm always kept at most equal or lower that those obtained by the conventional MD procedure. The fact that the outcomes of model selection mostly coincided for the classical and the VQSVM approaches witnesses the relative advantage of the latter method within the framework of the Structural Risk Minimization principle.

3.4.4 Conclusions

The search for the best-fitting model is a crucial issue in designing a learning machine. The solution of this problem depends on an accurate estimation of the generalization error or, at least, on the characterization of the trend of the error with respect to the configuration parameters. The Maximum-Discrepancy (MD) probabilistic method for assessing a classifier’s generalization ability features a sound theoretical background and can provide analyti-

Table 3.5: “Heart” testbed. Comparison between classical MD and VQSVM generalization bounds

$2\sigma^2$	C									
	0.01		0.1		1		10		100	
	$R^{(TG)}$	$R^{(VQ-TG)}$	$R^{(TG)}$	$R^{(VQ-TG)}$	$R^{(TG)}$	$R^{(VQ-TG)}$	$R^{(TG)}$	$R^{(VQ-TG)}$	$R^{(TG)}$	$R^{(VQ-TG)}$
1e-2	51.5%	51.5%	51.47%	51.42%	86.05%	75.42%	86.05%	74.7%	86.05%	71.97%
1e-1	51.47%	51.47%	51.47%	51.42%	86.05%	75.42%	86.05%	74.7%	86.05%	71.97%
1	50.77%	50.77%	50.77%	50.77%	86.05%	75.55%	86.05%	74.7%	86.05%	71.97%
10	35.75%	35.75%	35.75%	35.75%	79.4%	65.07%	86.05%	79.82%	86.05%	79.42%

Table 3.6: “Heart” testbed. Measured test error for model selection validation

$2\sigma^2$	C				
	0.01	0.1	1	10	100
1e-2	6.41%	6.41%	10.16%	10.16%	10.16%
1e-1	6.41%	6.41%	10.16%	10.16%	10.16%
1	6.41%	6.41%	10.16%	10.16%	10.16%
10	10.69%	10.69%	19.78%	21.92%	21.92%

Table 3.7: “Ionosphere” testbed. Comparison between classical MD and VQSVM generalization bounds

$2\sigma^2$	C									
	0.01		0.1		1		10		100	
	$R^{(TG)}$	$R^{(VQ-TG)}$	$R^{(TG)}$	$R^{(VQ-TG)}$	$R^{(TG)}$	$R^{(VQ-TG)}$	$R^{(TG)}$	$R^{(VQ-TG)}$	$R^{(TG)}$	$R^{(VQ-TG)}$
1e-2	43.06%	43.06%	43.06%	43.06%	100%	99.52%	100%	100%	100%	98.42%
1e-1	43.06%	43.06%	43.06%	43.06%	97.88%	96.57%	100%	97.76%	100%	97.96%
1	43.06%	43.06%	43.35%	43.24%	78.48%	72.49%	95.88%	76.15%	99.47%	73.99%
10	43.06%	43.06%	13.36%	12.53%	46.86%	40.58%	67.72%	54.62%	85.62%	66.91%

Table 3.8: “Ionosphere” testbed. Measured test error for model selection validation

$2\sigma^2$	C				
	0.01	0.1	1	10	100
1e-2	29%	29%	28%	28%	28%
1e-1	29%	29%	28%	28%	28%
1	29%	29%	11%	10%	10%
10	29%	5%	5%	3%	5%

Table 3.9: “Sonar” testbed. Comparison between classical MD and VQSVM generalization bounds

$2\sigma^2$	C									
	0.01		0.1		1		10		100	
	$R^{(TG)}$	$R^{(VQ-TG)}$	$R^{(TG)}$	$R^{(VQ-TG)}$	$R^{(TG)}$	$R^{(VQ-TG)}$	$R^{(TG)}$	$R^{(VQ-TG)}$	$R^{(TG)}$	$R^{(VQ-TG)}$
1e-2	57.06%	57.06%	57.06%	57.06%	100%	100%	100%	100%	100%	100%
1e-1	57.06%	57.06%	57.06%	57.06%	100%	100%	100%	100%	100%	100%
1	56.97%	56.97%	56.97%	56.95%	99.16%	98.88%	100%	99.89%	100%	99.89%
10	53.72%	53.72%	37.85%	37.85%	63.28%	62.04%	96.62%	78.62%	100%	78.41%

Table 3.10: “Diabetes” testbed. Comparison between classical MD and VQSVM generalization bounds

$2\sigma^2$	C									
	0.01		0.1		1		10		100	
	$R^{(TG)}$	$R^{(VQ-TG)}$	$R^{(TG)}$	$R^{(VQ-TG)}$	$R^{(TG)}$	$R^{(VQ-TG)}$	$R^{(TG)}$	$R^{(VQ-TG)}$	$R^{(TG)}$	$R^{(VQ-TG)}$
1e-2	46.64%	46.64%	46.64%	46.64%	100%	97.76%	100%	89.10%	100%	84.73%
1e-1	45.63%	45.63%	45.63%	45.63%	96.6%	91.83%	100%	95.92%	100%	96.36%
1	42%	41.58%	42.31%	42.06%	55.57%	49.84%	80.4%	74.81%	99.47%	89.88%
10	41.42%	41.07%	41.42%	41.06%	36.82%	34.46%	43.28%	41.44%	85.62%	50.02%

Table 3.11: “Diabetes” testbed. Measured test error for model selection validation

$2\sigma^2$	C				
	0.01	0.1	1	10	100
1e-2	34.75%	34.75%	34.75%	34.75%	34.75%
1e-1	34.75%	34.75%	34.57%	34.01%	34.01%
1	34.75%	34.75%	25.65%	26.95%	34.20%
10	34.75%	34.75%	25.65%	23.42%	24.72%

cal bounds [26]. Thus MD-based estimation provides the basic approach for assessing the run-time performance of Support Vector classifiers. A significant drawback of this method lies in the fact that, in real applications, the resulting MD estimates often does not well track the validation error.

To overcome this hindrance this study has proposed a criterion to identify and sort a subset of admissible functions within the considered general model. The method uses unsupervised learning to derive a ‘reference’ classifier, and the SVM parameters trained on the actual targets to set a limiting boundary. Only the SVM classifiers that lie within that boundary, with respect to the reference, are admissible when computing the generalization bound. The section has described an express procedure for implementing the optimization process under the constrained-capacity mechanism.

The effectiveness of the method has been illustrated by using real world datasets; results confirmed that the constraint-based approach leads to optimal hyper-parameters and featured tighter bounds than the conventional SVM method. Moreover, the predicted optimal models proved more accurate in terms of validation error, as compared with those obtained by applying the Maximal Discrepancy estimation to classical SVM’s when a cluster hypothesis exists and when the vector quantization algorithm proves effective in identifying the clusters structure.

In this work, the constraining method was applied to SVM by using VQ results as a reference. In fact, the methodology has a wider general validity, and the approach can extend to other classifier models. Since choosing a certain reference solution is equivalent to choosing a ‘prior’, other reference solutions than VQ can be adopted. The work presented here mainly aimed to set a starting point on the refinement of SRM, by using biased regularization for the computation of generalization bounds.

Next section will investigate the effect of applying biased regularization to various learning machines with the aim of obtaining a Semi-Supervised learning scheme: this means studying VQSVM and its extensions as a learning tool and no more only as a support tool for SVM tight bounds computation.

3.5 Semi-Supervised Learning by Biased Regularization

Inductive bias is of fundamental importance in learning theory, as it influences heavily the generalization ability of a learning system. From a mathematical point of view, the inductive bias can be formalized as the set of assumptions that determine the choice of a particular class of functions to support the learning process. Therefore, it represents a powerful tool to embed the prior knowledge on the applicative problem at hand.

This work addresses the advantages and the issues of introducing an inductive bias in kernel machines when semi-supervised classification problems are being tackled. In semi-supervised classification, one exploits both unlabeled and labeled data to learn a classification rule/function empirically; the semi supervised approach should improve over the classification rule that is learnt by only using labeled data.

The interest in semi-supervised learning has increased recently, especially because application domains exist (e.g., text mining, natural language processing, image and video retrieval, and bioinformatics), in which large datasets are available but labeling is difficult, expensive, or time consuming.

Biased regularization provides a viable approach to implement an inductive bias in a kernel machine, as confirmed by the generalized 'Representer Theorem' [48]. Biased regularization of Support Vector Machines (SVMs) has been adopted in [104] for a personalized handwritten system. Ivanov-like biased regularization was adopted in the previous section to shrink the generalization error bounds for SVMs under the so called *cluster hypothesis* (see [105] for a very in depth analysis). A similar result was obtained in the PAC Bayesian framework [106]. The research presented here shows that semi-supervised learning can benefit from biased regularization, too. First, a novel, general biased-regularization scheme is introduced that encompasses the biased versions of two powerful kernel machines, namely, SVMs and Regularized Least Squares (RLS). Then, the paper proposes a semi-supervised learning model, which is based on that biased-regularization scheme and follows a two-step procedure. In the first step, an unsupervised clustering of the whole dataset (including both labeled and unlabeled data) obtains a refer-

ence solution; in the second step, the clustering outcomes drive the learning process in a biased RLS (bRLS) machine or a biased SVM (bSVM) to acquire the class information provided by labels. The ultimate result is that the overall learned function exploits both labeled and unlabeled data. The integrated framework applies to both linear and non linear data distributions: in the former case, one works under a cluster assumption on data; in the latter case, one works under a manifold hypothesis [107]. As a consequence, for a successful semi-supervised learning, unlabeled data are assumed to carry some intrinsic geometric structure, e.g., in the ideal case, a low-dimensional, non-linear manifold.

The biased semi-supervised approach exhibits several features, such as modularity in the procedure that generates a biasing solution, convexity of the cost function, predictable complexity, and out-of-sample extension. Moreover, the paper shows that the framework allows one to perform model selection based on generalization bounds; this property proves especially useful in the presence of small labeled dataset (i.e., less than 50 patterns), and seems to represent an interesting novelty point in the scientific landscape of semi-supervised learning models. The experimental verification of the method involved both linear and non linear datasets. Experimental results confirmed the effectiveness of the approach for both bSVM and bRLS and proved that the proposed semi-supervised learning scheme compares positively with state-of-the-art algorithms, such as LapRLS, LapSVM [107], and Transductive SVM (TSVM) [3].¹

3.5.1 Biased Regularization

In the linear domain one can define a generic convex loss function, $l(X, Y, w)$, and a biased regularizing term; the resulting cost function is:

$$l(X, Y, \mathbf{w}) + \frac{\lambda_1}{2} \|\mathbf{w} - \lambda_2 \mathbf{w}_0\|^2 \quad (3.144)$$

where \mathbf{w}_0 is a “reference” hyper-plane, λ_1 is the classical regularization

¹ A Matlab implementation of the routines performing the comparison between the proposed framework, LapRLS, LapSVM and TSVM is available at http://www.sealab.dibe.unige.it/biased_learning

parameter that controls smoothness (e.g., $1/C$ in SVM), and λ_2 controls the adherence to the reference solution \mathbf{w}_0 . Expression (3.144) is a convex functional and thus admits a global solution. From (3.144) one gets:

$$l(X, Y, \mathbf{w}) + \frac{\lambda_1}{2} \|\mathbf{w} - \lambda_2 \mathbf{w}_0\|^2 = l(X, Y, \mathbf{w}) + \frac{\lambda_1}{2} \|\mathbf{w}\|^2 - \lambda_1 \lambda_2 \mathbf{w} \cdot \mathbf{w}_0 \quad (3.145)$$

which actually involves two regularization parameters, λ_1 and λ_2 ; this problem setting differs from the one proposed for SVM in [104], where only one regularization parameter was defined, obtaining $l(X, Y, \mathbf{w}) + \lambda_1 \|\mathbf{w} - \mathbf{w}_0\|^2$. The latter expression and (3.145) coincide in the special case $\lambda_2 = 1$.

Figure 3.20 (a,b,c) explicates the role played by parameter λ_2 in three different cases. In all those figures, a black square denotes the reference hyperplane, \mathbf{w}_0 , and a grey square indicates the 'true' optimal solution \mathbf{w}^* . For the sake of clarity, and without loss of generality, the examples assume that:

1. λ_1 is set to a fixed value (i.e., $\lambda_1 = 1$).
2. The distance $\|\mathbf{w} - \lambda_2 \mathbf{w}_0\|$ is constant for any λ_2 .
3. $\mathbf{w}_{\lambda_2=0}$ (black triangle) is the best solution one can obtain from the unbiased learning (i.e., $\lambda_2 = 0$). Here, 'best solution' refers to the solution that is closest to \mathbf{w}^* among all the possible \mathbf{w} that lies at a distance $\|\mathbf{w} - \lambda_2 \mathbf{w}_0\|$ from \mathbf{w}_0 (the dashed circumference).

Figure 3.20(a) refers to the situation in which the reference \mathbf{w}_0 is closer to the true solution \mathbf{w}^* than $\mathbf{w}_{\lambda_2=0}$. The Figure shows that when λ_2 decreases from 1 to 0, the centre of the ideal circumference, which encloses the eventual solution \mathbf{w}_{λ_2} , drifts. When $\lambda_2 \rightarrow 0$, \mathbf{w}_{λ_2} moves toward the origin $\mathbf{w} = 0$, which represents the condition 'no reference is exploited.' Indeed, the draw highlights that, when \mathbf{w}_0 gives a reliable reference, one can take full advantage of biased regularization, as the best solution for $\lambda_2 = 1$, $\mathbf{w}_{\lambda_2=1}$, definitely improves over $\mathbf{w}_{\lambda_2=0}$.

Figure 3.20 (b) illustrates the opposite case: the reference \mathbf{w}_0 is more distant from the true solution \mathbf{w}^* than $\mathbf{w}_{\lambda_2=0}$ (it is worth to note that the relative position of \mathbf{w}^* and $\mathbf{w}_{\lambda_2=0}$ with respect to the origin $\mathbf{w} = 0$ remains unchanged

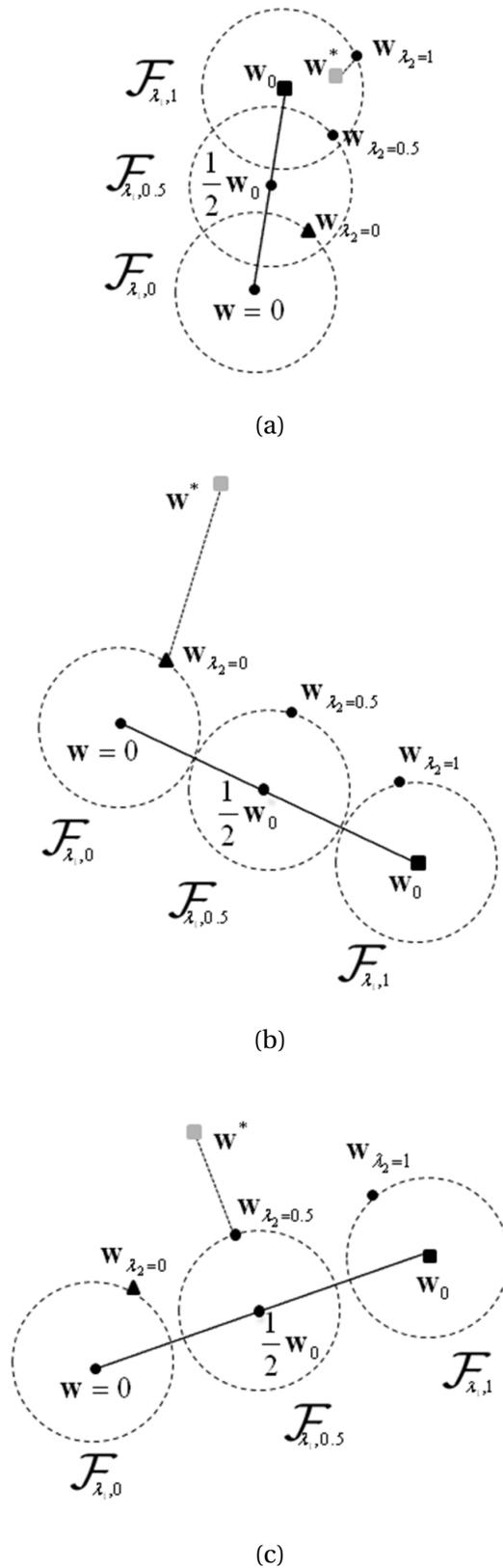


Figure 3.20: Role of λ_2 in biased regularization

when compared with Fig. 3.20(a)). In this situation, one would obtain the best outcome by setting $\lambda_2 = 0$, thus neutralizing the contribution of the biased regularization. Hence, \mathbf{w}_0 does not represent a helpful reference.

Finally, Fig. 3.20(c) illustrates another situation in which the reference \mathbf{w}_0 is more distant from the true solution, \mathbf{w}^* , than $\mathbf{w}_{\lambda_2=0}$. In such a peculiar situation biased regularization remains useful, as by adjusting λ_2 (i.e., by modulating the contribution of the reference \mathbf{w}_0) one eventually obtains a solution \mathbf{w}_{λ_2} that improves over $\mathbf{w}_{\lambda_2=0}$. As a result, one can take advantage of biased regularization even when the reference solution is not optimal.

The extension of (3.144) to non linear models is straightforward by considering a Reproducing Kernel Hilbert Space \mathcal{H} . In that case one has a reference function f_0 and the functional (3.144) becomes:

$$l(X, Y, f) + \frac{\lambda_1}{2} \|f - \lambda_2 f_0\|_{\mathcal{H}}^2, \quad (3.146)$$

where now the norm of the regularizer is taken in \mathcal{H} as usual. Eventually, one obtains the models for the biased RLS (bRLS) and the biased SVM (bSVM), respectively, by adopting the proper loss function $l(X, Y, \mathbf{w})$:

$$\begin{aligned} \text{bRLS} : & \quad \sum_{i=1}^l (y_i - f(x_i))^2 + \frac{\lambda_1}{2} \|f - \lambda_2 f_0\|_{\mathcal{H}}^2 \\ \text{bSVM} : & \quad \sum_{i=1}^l (1 - y_i f(x_i))_+ + \frac{\lambda_1}{2} \|f - \lambda_2 f_0\|_{\mathcal{H}}^2 \end{aligned}$$

This framework can be arbitrarily extended to n-layered networks. In the case of a two layered neural network, one should additionally constrain the hidden layer weights matrix: \mathbf{W}_0 denotes the reference of the hidden weights matrix and $\|\mathbf{W}\|_F^2$ is the Frobenius norm of matrix \mathbf{W} . Then a suitable functional to be minimized is:

$$l(X, Y, \mathbf{w}, \mathbf{W}) + \lambda_1 \|\mathbf{W} - \lambda_2 \mathbf{W}_0\|_F^2 + \frac{\lambda_1}{2} \|\mathbf{w} - \lambda_2 \mathbf{w}_0\|^2$$

In the following, the paper only considers networks such as kernel methods, where the hidden layer is not subject to empirical learning.

3.5.2 Biased SVM

The following Theorem formalizes the dual form of SVM that include a regularizing bias (bSVM). The formulations do not explicitly use the conventional scalar bias 'b' because it is implicitly considered by adding the value '1' at the end of each pattern.

Theorem 3.5.1. (bSVM): *Given a reference hyperplane \mathbf{w}_0 (or a reference function f_0), a regularization constant C , and biasing constant λ_2 , the dual form of the learning problem:*

$$\left\{ \begin{array}{l} \min_{\varepsilon, \mathbf{w}} C \sum_{i=1}^l \varepsilon_i + \frac{1}{2} \|\mathbf{w} - \lambda_2 \mathbf{w}_0\|^2 \\ y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 1 - \varepsilon_i \quad \forall i \\ \varepsilon_i \geq 0 \quad \forall i \end{array} \right. \quad (3.147)$$

is written as:

$$\left\{ \begin{array}{l} \min_{\alpha} \frac{1}{2} \alpha^t \mathbf{Q} \alpha - \sum_{i=1}^l \alpha_i (1 - \lambda_2 y_i f_0(x_i)) \\ 0 \leq \alpha_i \leq C \quad \forall i \end{array} \right. \quad (3.148)$$

The model of the data is:

$$f(x) = \sum_{i=1}^l \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + \lambda_2 f_0(\mathbf{x}) \quad (3.149)$$

Proof. See Appendix.

The minimization problem in (3.148) does not involve a linear constraint, hence it differs from the conventional SVM formulation [3]. The problem (3.148) can be optimized by an SMO version that uses only one Lagrange multiplier at each iteration. An *ad-hoc*, one-variable SMO algorithm can be easily derived from [108] or [104]. In such a new procedure, the gradient integrates the new reference based-term and the regularization parameter. The gradient value for the i -th pattern is:

$$G_i = y_i \sum_{j=1}^l \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_i) - 1 + \lambda_2 y_i f_0(\mathbf{x}_i) \quad (3.150)$$

Then, as usual, the projected gradient PG is computed and the KKT optimality conditions are checked on this value. The algorithm runs till the KKT conditions are satisfied. In the following the pseudo-code of the algorithm is presented.

Algorithm 6 bSVM Solver

Require: $\mathbf{Q}, \lambda_2, C, y, \tau$

Ensure: α

```

1:  $\alpha = 0, flag=0$ 
2: while  $!flag$  do
3:    $flag = 1$ 
4:   for  $i=1, \dots, l$  do
5:      $G = y_i \sum_{j=1}^l \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_i) - 1 + \lambda_2 y_i f_0(\mathbf{x}_i)$ 
6:      $PG = \begin{cases} \min(G, 0) & \text{if } \alpha_i = 0 \\ \max(G, 0) & \text{if } \alpha_i = C \\ G & \text{if } 0 < \alpha_i < C \end{cases}$ 
7:     if  $|PG| > \tau$  then
8:        $\alpha_i = \min(\max(\alpha_i - G/k_{ii}, 0), C)$ 
9:        $flag=0$ 
10:    end if
11:  end for
12: end while

```

The above pseudo-code can be considerably accelerated by updating the gradient only when necessary, and by using shrinking and random permutations of indexes of patterns at each iteration [108].

3.5.3 Biased RLS

The following theorem formalizes the linear biased version of RLS.

Theorem 3.5.2. *Given a reference hyperplane \mathbf{w}_0 , a regularization constant λ_1 , and a biasing constant λ_2 , the problem:*

$$\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda_1 \|\mathbf{w} - \lambda_2 \mathbf{w}_0\|^2$$

(3.151)

has solution:

$$\mathbf{w} = (\mathbf{X}^t \mathbf{X} + \lambda_1 \mathbf{I})^{-1} (\mathbf{X}^t \mathbf{y} + \lambda_1 \lambda_2 \mathbf{w}_0)$$

Proof. See Appendix.

The following theorem gives the dual form of biased RLS (bRLS):

Theorem 3.5.3. *Given a reference hyperplane \mathbf{w}_0 (or a reference function f_0), a regularization constant λ_1 , and a biasing constant λ_2 , the dual of the problem:*

$$\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda_1 \|\mathbf{w} - \lambda_2 \mathbf{w}_0\|^2$$

is:

$$\min_{\beta} \|\mathbf{K}\beta + \lambda_2 f_0(\mathbf{x}) - \mathbf{y}\|^2 + \lambda_1 \beta^t \mathbf{K} \beta$$

which has solution:

$$\beta = (\mathbf{K} + \lambda_1 \mathbf{I})^{-1} (\mathbf{y} - \lambda_2 f_0(\mathbf{X})) \quad (3.152)$$

The model of the data is:

$$f(\mathbf{x}) = \sum_{i=1}^l \beta_i K(\mathbf{x}, \mathbf{x}_i) + \lambda_2 f_0(\mathbf{x})$$

Proof. See Appendix.

From a computational point of view, the choice between the primal or the dual solution depends on the characteristics of the available data. If samples lie in a low-dimensional space and can be separated by a linear classifier, the primal form is preferable because it scales linearly with the number of features. Conversely, when the number of patterns is lower than the number of features, the dual form should be used.

3.5.4 Semi-Supervised Learning by using Biased Regularization

Kernel machines exploiting biased regularization can effectively support semi-supervised classification. This Section proposes the semi-supervised classification scheme, and discusses the appealing advantages provided by adopting the bSVM or the bRLS model as a learning machine in that framework.

3.5.4.1 A Semi-Supervised Learning Scheme Based on Biased Regularization

Let \mathbf{X} be a dataset composed by l labeled patterns and u unlabelled patterns; let \mathbf{X}_l denote the first subset, \mathbf{y} denote the corresponding vector of labels, and \mathbf{X}_u denote the second subset. Then the proposed semi-supervised learning scheme can be formalized as follows:

1. **Clustering:** Perform unsupervised clustering (bi-partite in the simplest case) of the dataset \mathbf{X} by adopting any algorithm supporting that task.
2. **Calibration:** For each cluster, set the cluster label by adopting a majority voting scheme that exploits the labeled samples. Then, for each cluster, assign to each sample the cluster label. Let $\hat{\mathbf{y}}$ denote this new set of labels.
3. **Mapping:** Given \mathbf{X} and $\hat{\mathbf{y}}$, train the selected learning machine and obtain the solution \mathbf{w}_0 .
4. **Biasing:** Given \mathbf{X}_l and the true labels \mathbf{y} train the biased version of learning machine (biased by \mathbf{w}_0). The eventual solution \mathbf{w} endows information from both the labeled data \mathbf{X}_l and the unlabeled data \mathbf{X}_u .

In this procedure, ‘Biasing’ step, is fully supported by the biased-regularization scheme previously introduced. The extension of the procedure to the non linear domain is straightforward: one should simply substitute \mathbf{w}_0 and \mathbf{w} with their kernel representations. The procedure has similarities to that adopted in deep learning architectures [109]. In that case the training algorithm performs a preliminary unsupervised stage, then uses labels only to adjust the network for the specific classification task; the eventual representation mostly reflects the outcome of the learning process completed in pre-training phase. Likewise, in the proposed framework, a pre-training phase builds \mathbf{w}_0 and a final adjustment derives the final \mathbf{w} . The semi-supervised learning scheme possesses some interesting features:

1. The learning scheme is general, as it applies both to linear and non linear domains. Any regularization based learning machine can be used:

neural networks with weight decays [38], random neural networks [70] plus regularization terms, kernel methods [1] and so on.

2. The semi-supervised learning task is tackled by separating the two actions: clustering, and biasing. One can control and adjust a specific action separately, e.g., by adopting a particular solution or by designing a new algorithm. This may be the case for the clustering task, which can take advantage from methodologies that address effectively complex, non linear domains.
3. If the learning machine is a single layer learning machine whose cost is convex then convexity is preserved and a global solution is granted.
4. Every clustering method can be used to build the reference solution.

It has been proved that simple optimization algorithms (bSVM) or standard linear system solvers (bRLS) can be used to run the learning algorithm for two powerful and reliable (biased) classification machines. Hence, bSVM and bRLS represent two consistent candidates for tackling biased learning in the semi-supervised classification framework. Indeed, the use of bSVM or bRLS can also lead to other interesting outcomes concerning the eventual computational complexity of the method and the model selection procedure. Those aspects are addressed in the following.

3.5.4.2 bSVM and bRLS for Semi-Supervised Learning: Computational Complexity

The semi-supervised learning scheme presented above involves three computationally intensive steps: clustering, mapping and biasing. The clustering task can be accomplished by several, different clustering algorithms, which in turn are characterized by different computational complexities. Thus, the complexity of this task will be denoted generically as O_C . In principle, though, some efficient and effective solutions to implement powerful clustering algorithms such as K-Means or Spectral Clustering are available [75].

In the second step, mapping, the time complexity is entirely determined by the learning machine applied to all the $l+u$ available samples. For RLS this

would mean a complexity of $O((l+u)^3)$, i.e. the solution of the usual system of linear equations. When adopting SVM as learning machine, one can exploit the SMO algorithm, which scales in between $O(l+u)$ and $O((l+u)^2)$ [101].

The third step, biasing, requires one either to solve a linear system or using an SMO-like algorithm. In both cases, one also need to pre-compute once the predictions of the reference model $f_0(\mathbf{x})$ for all the labeled patterns (with d -dimensional patterns the eventual cost is $O(ud)$). As a result, when bRLS is adopted as learning machine, the computational complexity is:

$$O_{bRLS} = O_C + O((l+u)^3) + O(ud) + O(l^3)$$

In O_{bRLS} the dominant terms are $O((l+u)^3)$ and possibly the complexity O_C associated to the clustering task. When instead bSVM is used, the computational complexity is:

$$O_{bSVM} = O_C + O((l+u)^2) + O(ud) + O(l^2)$$

where the dominant terms are $O((l+u)^2)$ and, again, the complexity O_C . In this case, one assumes that $O(ud) < O((l+u)^2)$; this is a reasonable hypothesis, except for those cases where the data lie in a highly dimensional space. Therefore, the complexity of the training procedure roughly scales with the same complexity of the original learning machine. SVM scales approximately as $O(l^2)$, and its semi-supervised version (bSVM) scales as $O((l+u)^2)$; a similar behavior characterizes RLS and bRLS. Some final considerations can be added to the discussion concerning computational complexity:

1. Provided that the clustering engine scales well, the proposed semi-supervised learning scheme is able to address large scale problems. In specific domains such as text mining where linearity and data sparsity can be exploited, adaptations [108] of the learning algorithm can lead to extremely fast learning algorithms, which are able to deal with hundreds of thousands of patterns in few seconds [108].
2. New, unseen test patterns can be managed effectively since the class assignment exploits the closed form function (3.149).

3.5.4.3 Biased Regularization for Semi-Supervised Learning Supports Effective Model Selection

The proposed semi-supervised learning framework extends the supervised classification scheme presented in the previous section (VQSVM). That work showed that, when a cluster hypothesis holds, using clustering to set a reference solution leads to a sharp reduction of the space of possible functions. Such result is noteworthy in that it eventually leads to tight generalization bounds. Tight generalization bounds are in turn a necessary condition for supporting an effective model selection. To extend the scheme of VQSVM to semi-supervised learning, the present framework involves both labeled and unlabeled data in the clustering step. Indeed, the effectiveness of model selection is improved by adopting a formulation of biased regularization that fully exploits parameters λ_1 and λ_2 . These quantities actually implement the model discussed in [110], which analyzed the implication of using a strong bias or a weak bias on the hypothesis space.

Figure 3.21 considers four cases, and analyzes the relative positions of the origin $w = 0$, the reference, w_0 and the true solution, w^* . Figure 3.21 (a) exemplifies the case “good reference / weak bias,” whereas Fig. (b) illustrates the case “good reference / strong bias.” In both situations, the reference solution w_0 is not far from the true solution w^* . However, in the first case, a weak bias is adopted, and the biasing step is allowed to explore a relatively wide portion of space around the reference (the lighter circumference, which represents the space of functions). Conversely, the second case refers to the use of a strong bias, hence a smaller portion of space is explored. Eventually, the area explored by adopting a strong bias does not include w^* .

Figure 3.21 (c) refers to the case “bad reference / weak bias.” The reference w_0 is quite distant from the true solution w^* ; by adopting a weak bias, though, one can still exploit biasing to reach w^* . Finally, Figure 3.21 (d) presents the case “bad reference / strong bias.” In this situation, by adopting a strong bias one restricts the space to be explored to a small region around w_0 . As a result, the proposed solution will be very distant from the true solution w^* . Overall, the four examples confirm that by modulating the biasing mecha-

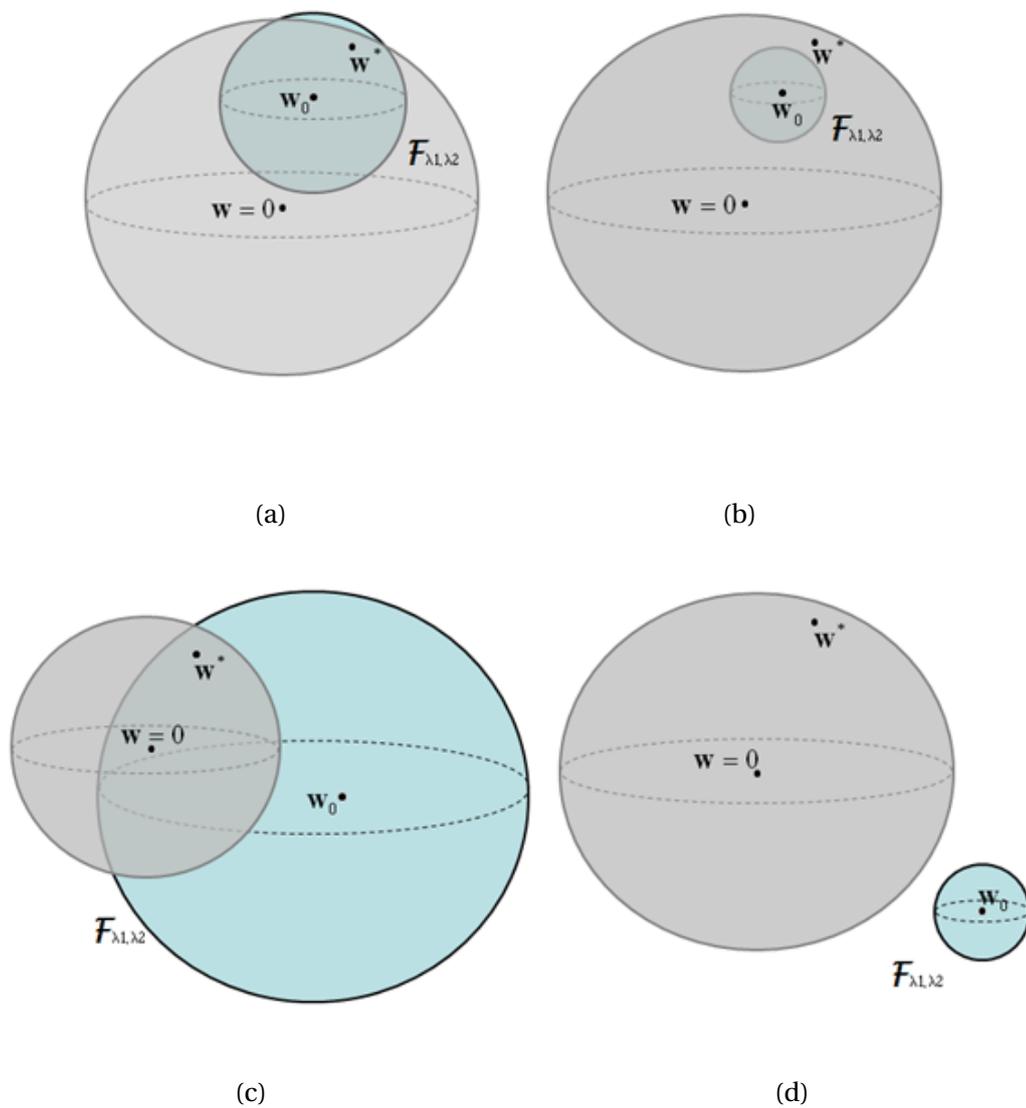


Figure 3.21: Inductive Bias and Hypothesis space

nism through the parameters λ_1 and λ_2 one can take full advantage of the semi-supervised scheme and support the model selection procedure properly. Eventually, the proposed framework involves a novel semi-supervised classification scheme, which supports a fully automated model selection and can be applied also when the size of the labeled dataset is small ($l < 50$). Such feature will be analyzed experimentally by exploiting the MD approach to assess true generalization errors.

3.5.4.4 Comparison with Other Semi-Supervised Classification Methods

A considerable number of semi-supervised clustering methods have been proposed in the past years [3][111][112][113][114]. Among them the most spread approaches include the co-training method [115], Transductive SVM (TSVM) [3], the semi-supervised generative model based on the Expectation Maximization (EM) algorithm [112], the cluster kernel method [113], manifold regularization [107], and Semi Parametric Regularization [114]. The proposed framework can provide attractive features when compared with these methods. First, the reference function can be worked out by exploiting any clustering algorithm. Such flexibility is not supported by the approaches discussed above, which in fact embed their own methodology for compute the reference function. Second, biased regularization can support effectively model selection, since tight generalization bounds can be attained [8]. This feature is crucial indeed, in particular when a model with few labeled data is addressed. Other approaches to semi-supervised learning do not provide this attribute. Furthermore, the present framework exploits a convex cost function. In fact, such feature is not supported by TSVM and the EM-based approach. As a major consequence, optimization represents a difficult task in both approaches. Actually, an efficient learning scheme for TSVM can be attained by exploiting Convex Concave Procedure [116]; analogously, some effective strategies are proposed in [112] to avoid local minima.

The proposed framework also ensures satisfactory performance in terms of computational complexity. Such aspect is highlighted in Table 3.12, which reports for each approach computational complexity, the implicit hypothesis made on data distribution in the feature space, the functional kind, and

Method	Complexity	Cluster Hypothesis	Functional	Generalization Bound
bRLS	$O_C + O(l + u)^3$	flexible	convex	yes
bSVM	$O_C + O(l + u)^2$	flexible	convex	yes
bSVM linear	$O_C + O(d(l + u)k)$	linear	convex	yes
bSVM linear sparse	$O_C + O(\bar{d}(l + u)k)$	linear	convex	yes
LapRLS	$O(l + u)^3$	manifold	convex	no
LapSVM	$O(l + u)^3$	manifold	convex	no
LapSVM linear	$O(d)^3$	linear	convex	no
Cluster kernel	$O(\max(l, u)^3)$	flexible	convex	no
TSVM	$O(k(l + 2u)^2)$	non dense region cuts	non convex	no
EM Based	-	mixture models	non convex	no
Co-training	-	-	classifier dependent	yes
Semi Parametric	$O(l + u)^3$	KPCA induced	convex	no

Table 3.12: Comparison among semi supervised classificatio methods

the availability of generalization bounds. Computational complexity is formalized by using the number of labeled patterns, l , the number of unlabelled patterns, u , the dimensionality of the data, d , the number of iterations of the learning algorithm, k , the complexity of the clustering algorithm, O_C . The Table also reports the computational complexity of bSVM linear under the hypothesis of sparse data; in that case, \bar{d} represents the average number of non null data. For the proposed biased learning machines, complexity has been formalized by assuming the use of efficient SMO-like routines. The Table shows that the present semi-supervised learning scheme can attain satisfactory performances in terms of computational complexity whenever the clustering algorithm scales as (or better than) the adopted biased machine. In this regard, bSVM appears especially appealing as it scales quadratically, or even linearly if the underlying problem has particular characteristics. Indeed, one should take into account that the term O_C represents the additional cost to be paid to gain in flexibility.

3.5.5 Experimental Results

The experimental section aimed at evaluating the accuracy performances of the proposed methods on unseen data, i.e. to assess induction performances. First, the results obtained on a toy dataset, two moons [117], are presented. Then, the performance of bSVM and bRLS are analyzed in a text mining problem. In particular, the ability of the proposed framework to support

effective model selection is showed. Finally, a comparison between bSVM, bRLS, LapRLS, LapSVM and TSVM is proposed.

3.5.5.1 Two-Moons Synthetic Dataset

The Two Moons dataset [117] includes 200 patterns, which lie on a bidimensional space. Figure 3.22 compares the results obtained on this synthetic dataset by exploiting two different semi-supervised classification approaches. Figure 3.22 (a) shows the classification rule learned by a conventional RLS machine fed with two labeled samples, one for each class; the two labeled patterns used in the experiment are marked in the figure. Obviously, RLS failed in finding out a satisfactory classification rule. Figure 3.22(b) reports the classification rule learned by adopting the proposed semi-supervised classification scheme; the same two patterns were used as labeled samples. In this case, bRLS was exploited as learning machine and linear k-means was adopted as clustering tool. Figure 3.22 (b) shows that the semi-supervised classification tool could not attain a reliable classification rule. Indeed, Fig. 3.22(c) gives the eventual result obtained by adopting a different configuration in the semi-supervised classification framework. In this set up, the clustering procedure was supported by Spectral Clustering, while bRLS still was used as learning machine; the regularization parameters were set as follows: $\lambda_1 = \lambda_2 = 1$ and $\sigma = 0.2$. The graph in Fig. 3.22 (c) proves that the semi-supervised framework was able to learn the correct classification rule when adopting the latter configuration.

Figure 3.22 (b) and 3(c) allow one to understand the crucial advantage provided by flexibility in the semi-supervised classification scheme. The unsatisfactory result illustrated in Fig. 3.22 (b) is caused by a poor performance of the clustering tool (linear k-means), which was not able to capture the non linear structure of the dataset. As a major consequence, the eventual reference solution could not support properly the classification scheme. Conversely, when a suitable clustering tool is exploited (in this example, Spectral Clustering) biased regularization can lead to effective results, even when only two labeled patterns are available (Fig 3.22 (c)).

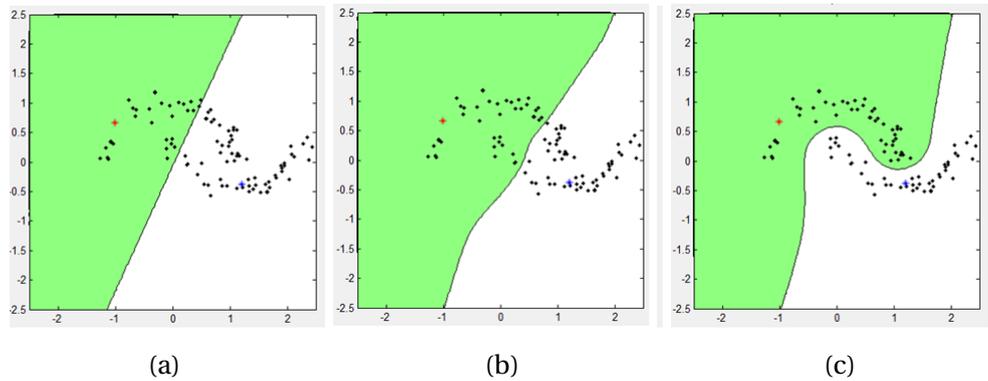


Figure 3.22: Two moons semi supervised learning

3.5.5.2 A Text Mining Problem

Text mining represents an important application domain for semi-supervised learning. Hence, the well known Newsgroup20 dataset was used as testbed to evaluate the performance of the proposed semi-supervised learning scheme on text mining problems. In the Newsgroup20 dataset text documents are represented as vectors of real-valued weight terms. First, each document undergoes a pre-processing phase, after which a document D eventually consists in a sequence of significant tokens called 'index terms.' The corresponding vector-space model spans a T -dimensional dictionary, and represents each document D as a vector. Each component of the T -dimensional vector is a non-negative weight term that denotes the relevance of the term itself within the document D (e.g., the term frequency). The patterns included in the Newsgroup20 dataset are partitioned across 20 different newsgroups, each corresponding to a specific topic. The present experimental session involved four binary classification problems: rec.autos vs rec.motorcycles, rec.sport.baseball vs rec.sport.hockey, sci.med vs sci.electronics and soc.religion.christian vs talk.politics.mideast. A pre-processing stage based on random projections [17] was applied to data to shrink the patterns dimensionality; eventually, the space dimension was reduced to 3000 features. The first experiment adopted conventional cross-validation to support the search for the best parameter setting. Four different learning machines were involved in the ex-

periment: SVM, RLS, semi-supervised learning by bRLS and semi-supervised learning by bSVM. The ranges for the model selection were set as follows: $C \in \{2^{-8}, 2^8\}$, $\lambda_1 \in \{2^{-8}, 2^8\}$ and $\lambda_2 \in \{2^{-8}, 2^8\}$. The primal form was adopted for the training of bRLS, while bSVM was trained using the dual form. The linear k-means algorithm was exploited as clustering tool in the semi-supervised learning scheme. That choice was motivated by the intrinsic linear structure of text mining problems. The experiment was conducted as follows. For each classification problem, 1000 randomly selected patterns were used for the training and cross-validation procedure; the remaining patterns (about 1000 samples) were collected in the validation set. Training and cross-validation involved different runs. In each run, the complete set of 1000 patterns composed the unlabeled dataset to be used in the clustering phase; conversely, only a subset of these patterns was considered labeled and thus available for the training phase. The labeled samples not involved in the training were indeed used as cross-validation set. Hence, eventually, different runs were developed by increasing progressively the number of labeled data involved in the training phase. The results are presented in Figure 3.23: each graph compares, for the single classification problem, the performance of the four learning machines involved in the experiment; the graph gives on the y-axis the classification error on the validation set and on the x-axis the number of labeled patterns used in the training phase. Figure 3.23 proves that the proposed semi-supervised classification scheme improves over powerful learning techniques such as RLS and SVM. In particular, the classification error dramatically decreases when a very small number of labeled patterns is available for training. Nonetheless, the semi-supervised scheme attains remarkable performance even when more than 500 labeled patterns are available.

The second experiment was designed to test the ability of the semi-supervised learning to perform effective model selection by exploiting generalization error bounds. Therefore, the best model was selected by choosing the parameter setting that led to the best generalization error, which was assessed by adopting the MD approach (details on the computation of generalization bounds are given in [8]). As before, the experiment involved different runs, corresponding to different sizes of the labeled dataset in the training phase.

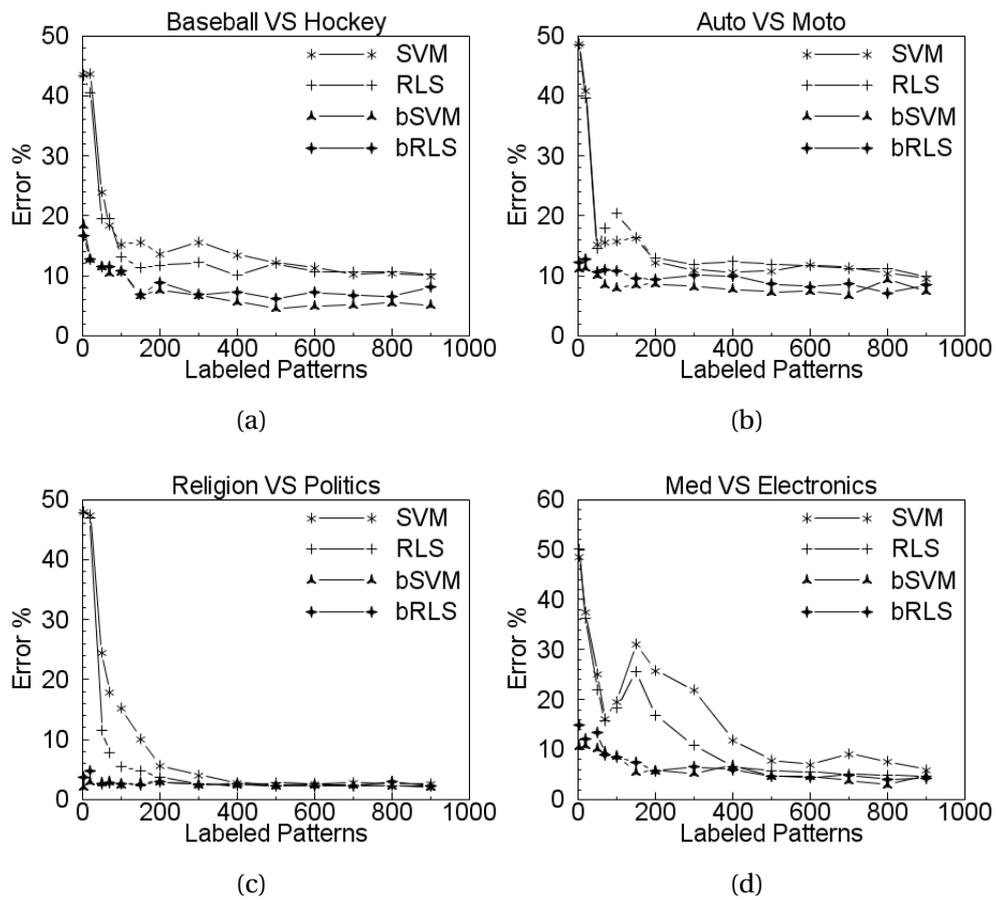


Figure 3.23: Semi-supervised text mining

Figure 3.24 presents the results obtained with the experiment. Each graph compares, for the single classification problem, the performance of SVM with that attained by the semi-supervised scheme exploiting bSVM; the reference performance obtained by bSVM with the conventional cross-validation procedure is also given. The graph gives on the y-axis the classification error on the validation set and on the x-axis the number of labeled patterns used in the training phase. In this experiment the range of labeled samples has been limited to 50, as conventional model selection based on cross validation is unreliable in particular when very small dataset are involved. Results confirm that the proposed semi-supervised scheme supports a reliable model selection procedure based on generalization bounds. The eventual classification error of the selected model is obviously worse than that attained with conventional model selection. However, the performance is still satisfactory.

3.5.5.3 Comparison with LapRLS, LapSVM and TSVM

The last experimental session was designed to compare the proposed semi-supervised scheme with other semi-supervised framework proposed in the literature: LapRLS, LapSVM [107], and Transductive SVM (TSVM) [3]. To perform a fair comparison, the experiments involved the datasets already addressed by those approaches: USPS [118] and Isolet [100]. Different configurations of the semi-supervised scheme were compared. In particular, four different clustering tools were used: k-means with Manhattan distance, k-means with Euclidean distance, spectral clustering, and Expectation Maximization. As publicly available Matlab code is provided for LapRLS, LapSVM and TSVM [117], experiments involving the proposed semi-supervised learning scheme were developed by embedding the new code into those routines. As a result, the experiments also exploited the data preprocessing designed by the authors of those approaches. To this aim, the conventional Matlab routine of the k-means algorithm was used, while publicly available Matlab versions of the EM algorithm [119] and of spectral clustering [75] were exploited. The corresponding Matlab code is freely available at: http://www.sealab.dibe.unige.it/biased_learning. The first experiment addressed the USPS dataset, which is a OCR dataset collecting digits images.

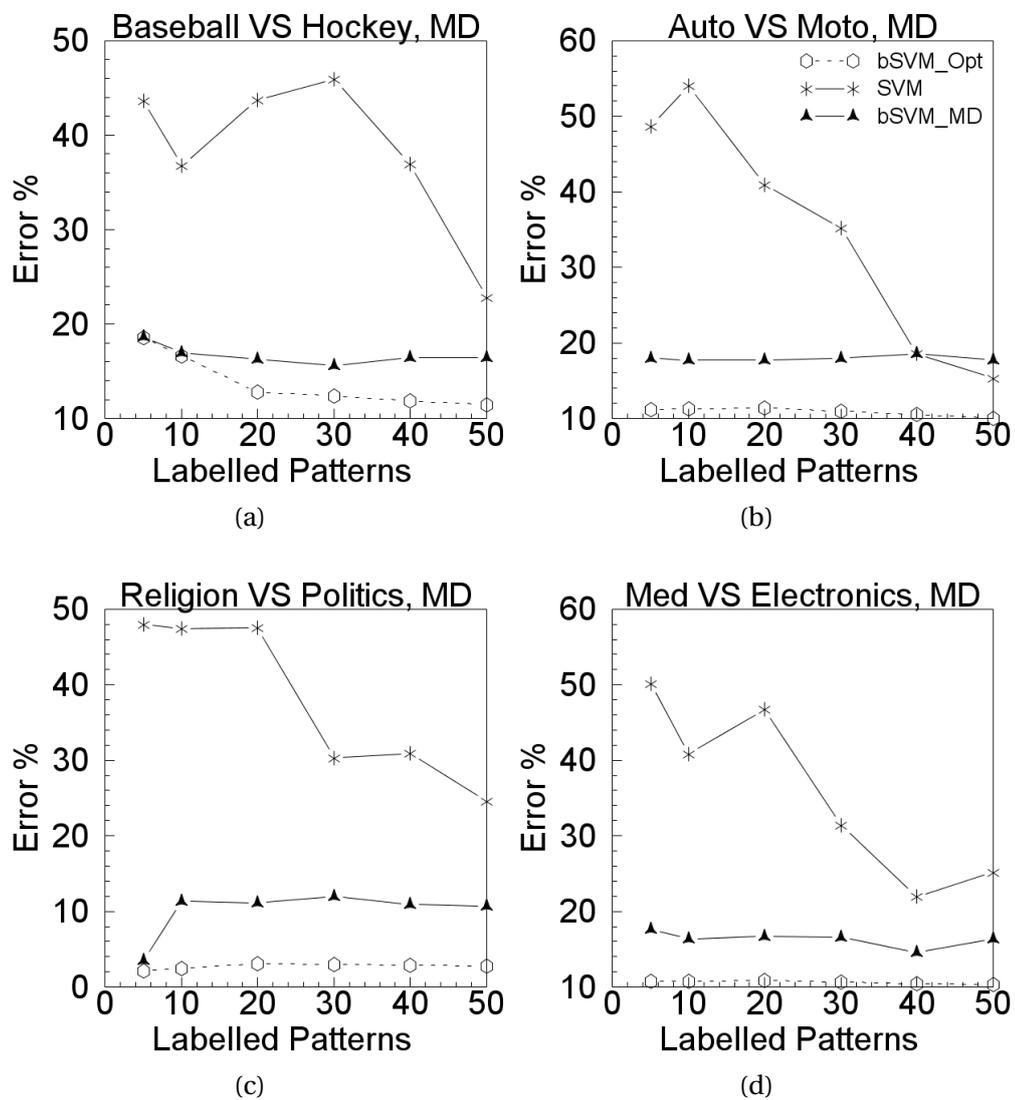


Figure 3.24: Semi-supervised MD text mining

The experiment involved all the 45 bi-class problems that can be generated from the dataset. For each problem, the experimental set up followed that adopted in [107]: the first 400 images are inserted in the training set and pre-processed by PCA, which is exploited to obtain a feature space with dimension 100; the remaining images compose the test set. For each class, 2 samples randomly selected were labeled, while the others were left unlabeled. Polynomial kernel was used of degree ‘3’. The parameters setting in [107] denotes that geometry information is very important to correctly address this problem, as a 9:1 ratio in the regularizers settings results in a predominance of the geometric term induced by the Graph Laplacian. Therefore, parameters λ_1 and λ_2 were set accordingly in all the experiments involving Spectral Clustering (whose solution is induced by the Graph Laplacian): $\lambda_1 = 0.1$ (i.e., $C = 10$) and $\lambda_2 = 1$. These settings result in a configuration that gives high confidence to the reference solution provided by clustering. Different settings were used in the experiments not involving Spectral Clustering: $\lambda_1 = 0.1$ (i.e., $C = 10$), $\lambda_2 = 0.1$; this configuration smoothes the importance of the reference solution. Such set up was the result of preliminary experiments on the dataset.

Figure 3.25 presents the comparison between the eight different configurations adopted for the proposed semi-supervised learning scheme. Fig. 3.25 (a) refers to the four configurations based on bSVM: the y-axis gives the classification error, while the x-axis enumerates the different classification problems involved in the experiment. Analogously, fig. 3.25 (b) presents the results obtained with the four configurations based on bRLS. The plots clearly show that the configurations that attain the best performance are those exploiting spectral clustering as clustering tool.

Figure 3.26 compares the performances of the two configurations based on spectral clustering with those attained by LapRLS, LapSVM and TSVM. In particular, fig. 3.26(a) gives the results attained by bRLS and LapRLS and fig. 3.26(b) gives the results attained by bSVM, LapSVM and TSVM. All the results refer to the accuracy values obtainable at the break-even points in the precision-recall curves; such set up follows the one adopted in [107] and allows a fair comparison between the different approaches. Numerical results

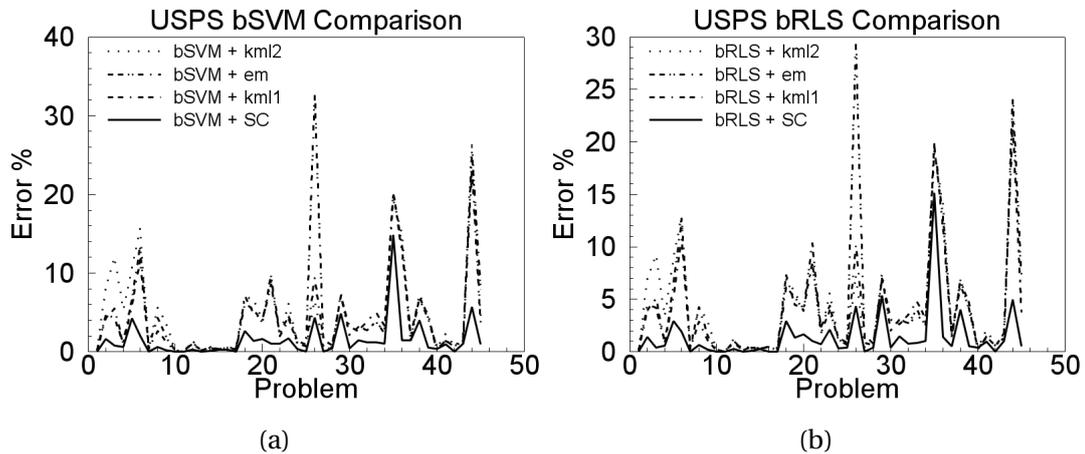


Figure 3.25: USPS Comparison with different reference clusterings

show that most of the time the proposed semi-supervised scheme improves over the other methods. Indeed, in some cases the gain in classification error obtained with bRLS or bSVM is significant.

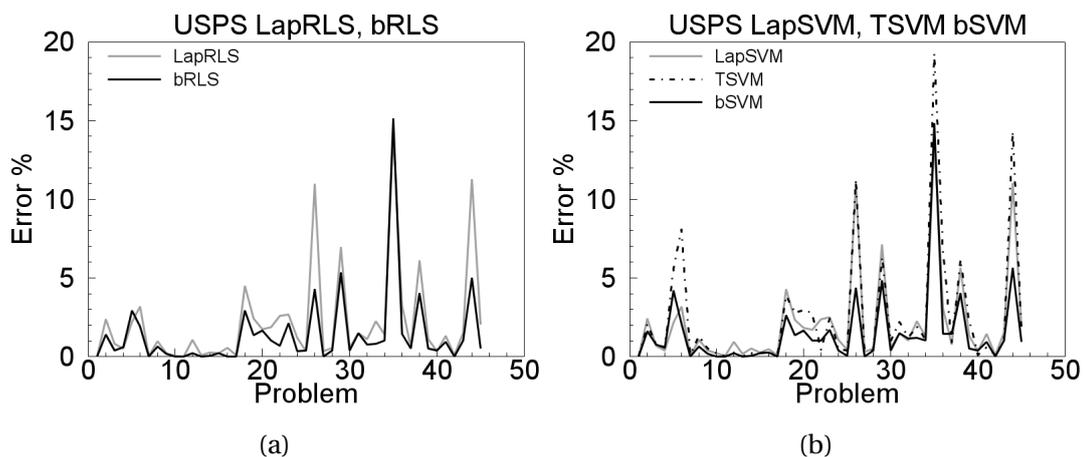


Figure 3.26: USPS Comparison with LapRLS, LapSVM, TSVM

The second experiment addresses the Isolet dataset; this dataset collects letters of the English alphabet spoken in isolation by 150 subjects [100]. The speakers are grouped in 5 sets of 30 speakers each (isolet1, isolet2, etc). Also in this case the experimental design followed the one adopted in [100]. Thirty speakers (the subset isolet1) are used for training, thus obtaining 1506 train-

ing patterns; the test set is composed by the isolet5 subset, which contains 1559 examples. The binary classification task consists in dividing the first 13 letters of the alphabet from the last 13. A total of 30 classifications problems are considered; each problem corresponds to a different data split where 52 utterances of one speaker were labeled, while the others remain unlabeled. The RBF kernel was exploited in this experiment; the kernel width σ was set to 10. In [107] the geometric regularization reference term was chosen to be one order of magnitude smaller than the first regularization term. Such setting clearly indicates that the confidence in the geometric information is low; this in turn suggests that semi-supervised learning may not be effective. Indeed, inductive results in [107] show a marginal gain over the conventional supervised solution.

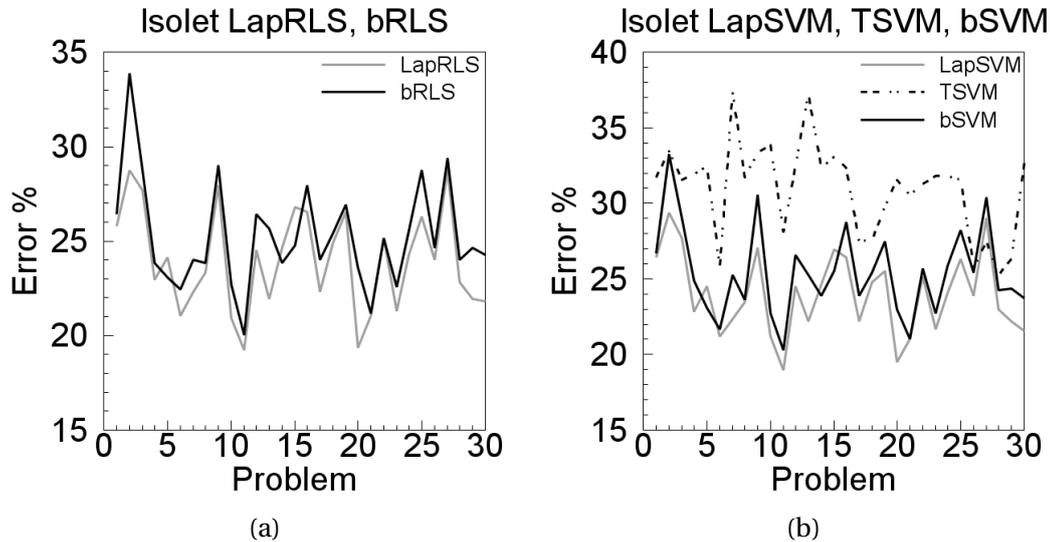


Figure 3.27: Isolet Comparison with LapRLS, LapSVM, TSVM

Figure 8 compares the performance of LapRLS, LapSVM, and TSVM with those attained by the proposed semi-supervised learning scheme (using spectral clustering as clustering tool). The regularization parameters were set as follows: $\lambda_1 = 0.1$ and $\lambda_2 = 0.1$. Also in this case the results refer to the accuracy values obtainable at the break-even points in the precision-recall curves. The plots show that the proposed semi-supervised learning method does not seem able to attain better performance than LapRLS or LapSVM for

this dataset. An analysis of the underlying data structure can indeed give an explanation for that outcome. The Spectral Clustering procedure seems unable to achieve a satisfactory division of patterns that actually lies in different classes. As a major consequence, the semi-supervised learning scheme suffers from such ineffective clustering. Better performances could be obtained with a clustering algorithm able to perform a different projection (if any); e.g., one may change the affinity matrix in the Spectral Clustering procedure to get a different mapping of the data.

3.5.6 Conclusions

The present research proves that using biasing techniques in kernel machines can lead to an effective, yet simple learning scheme for semi-supervised classification. The eventual framework is characterized by several appealing features. First, the semi-supervised learning task is tackled by separating the two actions: clustering, and biasing. Therefore, one can control and adjust a specific action separately, e.g., by adopting a particular solution or by designing a new algorithm. This may be the case for the clustering task, which can take advantage from methodologies that address effectively complex, non linear domains. Second, if the employed learning machine supports a convex optimization problem, the learning scheme preserves convexity. Moreover, the present semi-supervised learning scheme can support an accurate estimation of the generalization bounds; as a major result, effective model selection can be achieved even when small dataset are involved. The proposed framework also performs effectively in terms of computational complexity. In principle, the overall computational cost is affected by the additional cost brought about by clustering to be paid to gain in flexibility. Indeed, that additional term may be neglected provided that the clustering engine scales well with the number of samples. Two future activities can stem from the present work. The first addresses the use of biased regularization in other kernel machines or even in neural networks. The second aims at developing new biasing schemes that may improve the generalization ability of regularized learning machines.

3.6 Explicit Transductive bound

3.6.1 Induction, Transduction and Semi-Supervised learning

In recent years, approaches alternatives to full induction have reached an always increasing attention from the machine learning research community. Inductive methods find a global solution from empirical data and build general model applicable all over the population.

Beside inductive learning schemes, exist the so called transductive learning [3]: in this environment is not required generalization for every possible input, instead only achieving the best possible performance on a particular and known test data. This, intuitively, makes transduction simpler than induction, since what is request are values at given points [3] and not a global predictive function.

Transduction and semi-supervised learning are quite different concepts: in the first setting one is interested in finding values at given points and no more, in semi-supervised learning one is interested in producing a decision function by using labelled and unlabelled data: a transductive algorithm can perform predictions only on working set, a semi-supervised one can predict all over the population so it is completely inductive.

The importance of transduction is due to different reasons: one of them is its fundamental part over the inductive approach itself. Well known Vapnik classical bound, somehow implicitly makes use of transduction when concerned with ghost set. In transductive setting, the ghost set is real and is the set of given points in which predictions are performed. Another reason stems from the possibility to take advantage of this new simpler setting to get tighter bounds on generalization error over a particular working set. In this work this second aspect will be studied: adapting the machinery of Theorem 4.2 [3] and a relatively recent result [120] an explicit formula will be obtained for overall risk minimization bound.

In the first part of this section computational issues will be discussed over the numerical evaluation of transductive bound in its implicit original form and a closed form formula will be obtained; in the second one the result will

be compared to other existing bounds. The same symbolic conventions of [3] will be used throughout this work:

- $l + k$ is the total number of patterns; l are labelled and k are unlabelled
- ν_τ is the transductive error (the error over test or working sample); ν is the error on training set; ν_2 is the error on the ghost set; than we call $\nu_0 = \nu \frac{l}{l+k} + \nu_\tau \frac{k}{l+k}$, and $\nu_\alpha = \frac{\nu + \nu_2}{2}$
- m is the total number of errors and can be expressed as $\nu l + \nu_\tau k$
- $G(l+k)$ is the Grow Function computed for $l+k$; $H_{ann}^\Lambda(2l)$ is the annealed Entropy
- $1 - \delta$ is the confidence level of the bound
- C_m^r is the binomial coefficient
- $\Gamma_{l,k}(\varepsilon, m)$ is a quantity derived from the hypergeometric distribution; than we call $\Gamma_{l,k}(\varepsilon) = \max_m \Gamma_{l,k}(\varepsilon, m)$; $E(\Gamma)$ is the expectation of the hypergeometric; N_{l+k} is the finite number of equivalence classes
- the function space \mathcal{F} is composed by the functions $f(\mathbf{x}, \alpha)$ parametrized by α parameters that belong to the space of parameters Λ .

3.6.2 Overall Risk Minimization

The Overall Risk Minimization framework is part of the more general Statistical Learning Theory and is one of the possible approaches to transduction. In [3] transduction is introduced and two possible settings (Setting 1 and Setting 2) are exposed: it can be shown that both of them are equivalent [3]. The fundamental idea, on which ORM is built up, is that is useless solving a more difficult problem when a simpler one is needed to be solved. From a mathematical point of view in ORM we are endowed with a training labeled set, an unlabelled working set on which one wants to perform predictions and it is allowed to use both of them during training. This fundamental theorem gives an implicit bound for transduction error

Theorem 3.6.1. (Theorem 8.2 in [3]) Let the set of decision rules $f(\mathbf{x}, \alpha)$, $\alpha \in \Lambda$ on the complete set of vectors have N_{l+k} equivalence classes. Then the probability that the relative size of deviation for at least one rule in $f(x, \alpha)$, $\alpha \in \Lambda$ exceeds ε is bounded by:

$$P \left\{ \sup_{\alpha \in \Lambda} \frac{|\nu - \nu_\tau|}{\sqrt{\nu_0}} > \varepsilon \right\} < N_{l+k} \Gamma(\varepsilon) \quad (3.153)$$

Now, as said in the introductory section, one wants to link the equipment of induction to Theorem 8.2 [3]: in Statistical Learning Theory the main inductive result is Theorem 4.1 [3]; the key part in which one now is interested in, is Lemma 4.2 [3].

(Lemma 4.2 in [3]) For any $l > \varepsilon^{-2}$ is valid the following bound:

$$P \left\{ \sup_{\alpha \in \Lambda} \frac{|\nu_2 - \nu_1|}{\sqrt{\nu_\alpha + 1/(2l)}} > \varepsilon \right\} < \exp \left\{ H_{ann}^\Lambda(2l) - \frac{\varepsilon^2 l}{4} \right\} \quad (3.154)$$

further by using the property $H_{ann}^\Lambda(2l) < G(2l)$ for right hand side one gets $\exp \left\{ G(2l) - \frac{\varepsilon^2 l}{4} \right\}$. To make ORM approach consistent with machinery of Lemma 4.2 one needs to replace original Vapnik gamma function argument $\varepsilon \sqrt{\frac{m}{l+k}}$ with $\varepsilon \sqrt{\frac{m+1}{l+k}}$ (as in Lemma 4.2 [3] proof where one had $\varepsilon \sqrt{\frac{m+1}{2l}}$). This marginal modification leads also to modify (3.153) into:

$$P \left\{ \sup_{\alpha \in \Lambda} \frac{|\nu - \nu_\tau|}{\sqrt{\nu_0 + \frac{1}{l+k}}} > \varepsilon \right\} < N_{l+k} \Gamma(\varepsilon) \quad (3.155)$$

and the final explicit formula becomes:

$$\nu_\tau \leq \nu + \frac{k\varepsilon^2}{2(l+k)} + \varepsilon \sqrt{\left[\frac{k\varepsilon}{2(l+k)} \right]^2 + \nu + \frac{1}{l+k}} \quad (3.156)$$

This modification makes Theorem 8.2 [3] consistent with Lemma 4.2 [3] at price of adding the term $1/(l+k)$ over the original formulation; as will be seen later this adaptation open the possibility to build up a plain proof that makes ε term explicit.

After this simple variation one can try to appraise the implicit bound derived in Theorem 8.2 [3]. For its evaluation one has to find the smallest solution of $\ln N_{l+k} + \ln \Gamma(\varepsilon) < \ln \delta$ and plug it into the bound; so one has to solve

this equation for trials performing discretization on ε . Before proceeding it is necessary to explicitly compute the number of equivalence classes: for this purpose one can use the fact that $\ln N_{l+k} < G(l+k)$. This approach presents some performance problems when the number of patterns (l or k) is over $1e3$. The main issue consists in the explicit evaluation of the gamma function: its calculation plans the evaluation of three binomial coefficients. This function was implemented using Stirling and Ramanujan formulas and logarithmic representations, but despite this, the execution time is quite high when dealing with data mining problems.

As suggested by Vapnik itself, gamma function can be tabulated but it should be preferable having a simpler and explicit way of computing the bound. Although these concerns, evaluation of the bound has been possible via iterative search of the solution. For the exposed reasons a more practical solution consists in deriving an explicit bound.

3.6.3 Bound derivation

The subsequent theorem is the central result of this section: it follows Vapnik demonstration for the classical inductive bound (Theorem 4.2 [3], Lemma 4.2 [3]) and readapts it to the transductive issue (Theorem 8.2 [3]) using a quite recent statistical result [120].

Theorem 3.6.2. (*Explicit bound*). Assured that $G(l+k) - \ln \delta > 6$, and setting $\varepsilon^2 = \frac{G(l+k) - \ln \delta + 2}{2(lk)^2} (l+k)^3$, with probability $1 - \delta$, the bound in (3.156) is valid.

Proof. Suppose having a population of $l+k$ patterns in which there are m misclassified patterns. One selects randomly l of them. The probability that among the selected patterns there are r errors equals $\frac{C_m^r C_{l+k-m}^{l-r}}{C_{l+k}^l}$. The probability that the frequency of misclassified patterns in the first group (l) deviates from the frequency of errors in the second group (k) by the amount exceeding $\bar{\varepsilon}$ equals:

$$P \left\{ \left| \frac{r}{l} - \frac{m-r}{k} \right| > \bar{\varepsilon} \right\} = \sum_r \frac{C_m^r C_{l+k-m}^{l-r}}{C_{l+k}^l} = \Gamma_{l,k}(\bar{\varepsilon}, m) \quad (3.157)$$

Where the sum is taken over the value of r such that:

$$\max(0, m-k) \leq r \leq \min(l, m), \left| r - \frac{lm}{l+k} \right| > \bar{\varepsilon} \frac{lk}{l+k} \quad (3.158)$$

Note that both sides are always greater than 0. From [120] is known that: if

$$r - E(\Gamma) \geq 2 \quad (3.159)$$

is true, than:

$$\ln \Gamma_{l,k}(\bar{\varepsilon}, m) < -2\alpha((r - E(\Gamma))^2 - 1) \quad (3.160)$$

where $\alpha = \max\left(\frac{1}{l+1} + \frac{1}{k+1}, \frac{1}{m+1} + \frac{1}{l+k-m+1}\right)$. Knowing that $E(\Gamma) = \frac{ml}{l+k}$ one gets:

$$\Gamma_{l,k}(\bar{\varepsilon}, m) < \exp\left(-2\alpha\left(r - \frac{ml}{l+k}\right)^2 - 1\right) \quad (3.161)$$

Expressing $\bar{\varepsilon} = \varepsilon\sqrt{\frac{m+1}{l+k}}$ one gets: $\left|r - \frac{ml}{l+k}\right| > \varepsilon\frac{lk}{l+k}\sqrt{\frac{m+1}{l+k}}$. Note that because ones needs the square in (3.161), by using (3.158) $\left(r - \frac{lm}{l+k}\right)^2 > \left(\bar{\varepsilon}\frac{lk}{l+k}\right)^2$ holds. For proceeding one has to assure that the hypothesis on hypergeometric bound (3.159) and (3.158) both hold. A simple way for achieving this goal is to request that: $\left(r - \frac{lm}{l+k}\right)^2 > \max\left(\left(\bar{\varepsilon}\frac{lk}{l+k}\right)^2, 2^2\right)$. Now observe that asking $\bar{\varepsilon}\frac{lk}{l+k} > 2$ is a sufficient condition to refer to the only $\left(r - \frac{lm}{l+k}\right)^2 > \left(\bar{\varepsilon}\frac{lk}{l+k}\right)^2$ original condition; in this way, at the end of the proof, one will have to check for what values the expression $\bar{\varepsilon}\frac{lk}{l+k} > 2$ is true. With these hypothesis one can bound the hypergeometric on (3.155) getting:

$P\left\{\sup_{\alpha \in \Lambda} \frac{|\nu - \nu_\tau|}{\sqrt{\nu_0 + \frac{1}{l+k}}} > \varepsilon\right\} < N_{l+k} \max_m \left\{\exp\left\{-2\alpha\left(\left(r - \frac{ml}{l+k}\right)^2 - 1\right)\right\}\right\}$ than one gets:

$$P\left\{\sup_{\alpha \in \Lambda} \frac{|\nu - \nu_\tau|}{\sqrt{\nu_0 + \frac{1}{l+k}}} > \varepsilon\right\} < N_{l+k} \max_m \left\{\exp\left\{-2\alpha\left(\left(\varepsilon\frac{lk}{l+k}\sqrt{\frac{m+1}{l+k}}\right)^2 - 1\right)\right\}\right\} \quad (3.162)$$

It can be easily shown that hypergeometric dependent part of the previous formula is maximized for $m = 0$ (as happens in Vapnik inductive proof). This fact ($m = 0$) makes $\alpha = \max\left(\frac{1}{l+1} + \frac{1}{k+1}, 1 + \frac{1}{l+k+1}\right)$, that is equivalent to assert that $\alpha > 1$; for this reason one can replace α with 1. Observe that this operation on α slightly affects the quality of the bound, in facts in almost all real world problems (e.g $l, k > 10$) $\alpha \simeq 1$ holds.

Setting $m = 0$ and recalling that $\ln N_{l+k} < G(l+k)$ one obtains:

$$P\left\{\sup_{\alpha \in \Lambda} \frac{|\nu - \nu_\tau|}{\sqrt{\nu_0 + \frac{1}{l+k}}} > \varepsilon\right\} < \exp\left(G(l+k) - 2\left(\varepsilon^2 \frac{(lk)^2}{(l+k)^3} - 1\right)\right) \quad (3.163)$$

Remember that the right part of the above inequality is δ . So expressing all in terms of ε^2 , one gets: $\varepsilon^2 = \frac{G(l+k) - \ln \delta + 2}{2(lk)^2} (l+k)^3$. Pluggin this formula in (3.156) one gets the final expression of the bound where ε^2 is explicit.

Finally one has to check the correctness of $\bar{\varepsilon} \frac{lk}{l+k} > 2$ hypothesis. In other terms one has to verify that $\varepsilon^2 > 4 \frac{(l+k)^3}{(lk)^2}$. Then $\varepsilon^2 = \frac{G(l+k) - \ln \delta + 2}{2(lk)^2} (l+k)^3 > 4 \frac{(l+k)^3}{(lk)^2}$ that produces the condition of the theorem: $G(l+k) - \ln \delta > 6$.

3.6.4 Valuation and experimental results

The obtained result is valid when the Grow function is explicitly known. If Grow Function is not exactly known, Sauer lemma can be used to get a bound in terms of Vapnik-Chervonenkis dimension d_{vc} that leads to:

$$\nu_\tau \leq \nu + \frac{\beta}{4(l^2k)} (l+k)^2 + \sqrt{\frac{\beta}{2(lk)^2} (l+k)^3} \sqrt{\nu + k^2 \frac{\beta}{8(lk)^2} (l+k)} \quad (3.164)$$

where $\beta = d_{vc} \left(1 + \ln \frac{l+k}{d_{vc}}\right) - \ln \delta + 2$.

Note also that a when typical confidence value of .95 is used, the theorem hypothesis is $G(l+k) > 3$, that is very likely to happen in practice.

There are others aspects that need analysis: first of all it is appropriate to observe that for $k = l$ the bound becomes:

$$\nu_\tau \leq \nu + \frac{G(2l) - \ln \delta + 2}{l} + 2\sqrt{\frac{G(2l) - \ln \delta + 2}{l}} \sqrt{\nu + \frac{G(2l) - \ln \delta + 2}{4l}} \quad (3.165)$$

From a cognitive and mathematical point of view keeping $k = l$ and requesting l, k big enough makes the above bound quite similar to original Vapnik inductive bound; these bounds became very similar when the Grow Function is far less, in absolute value, than the number of patterns (e.g. this can happen in clustering based classifiers). For completeness of information original Vapnik formulation was:

$$\pi \leq \nu + 2\frac{G(2l) - \ln \delta + 2 \ln 2}{l} + 2\sqrt{\frac{G(2l) - \ln \delta + 2 \ln 2}{l}} \sqrt{\nu + \frac{G(2l) - \ln \delta + 2 \ln 2}{l}} \quad (3.166)$$

It is important to note down that in this case ($k = l$) the obtained bound is always convenient over induction (see 3.28). Roughly speaking explicit transduction bound is convenient over induction in this case because one did not

pay the price of the ghost set, because ghost set in this setting exists and it is represented by k patterns. When k and l are unbalanced this advantage is lost due to the behaviour of the hypergeometric distribution. One can verify [11] that the obtained results are quite loose; however a tighter version of this bound exists [121] due to a refined measure of concentration of the hypergeometric distribution; moreover rademacher complexity can be used to build a transductive bound [122]

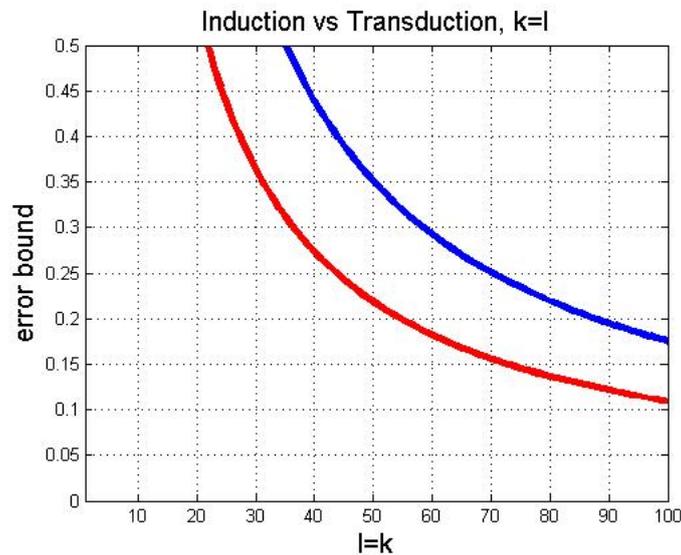


Figure 3.28: Experiment for $k = l$ variable. Note the advantage of transductive bound (red line) over induction (blue line)

Here a possible approach for building an explicit and simple to use transductive bound by using only Vapnik theory and without requiring any bayesian approach was presented. This result is far from the best possible transductive bound, instead the aim is to underline the critical role of the hypergeometric distribution on the Vapnik approach.

4

Applications

This chapter collects an etherogenous group of findings and applications of kernel methods and unsupervised learning in different applicative domains. A first work deals with text clustering methods: the discussed framework has been presented/used in several works

[17],[18],[19],[20],[21],[22] and it is now under the name of SLAIR, acronym of *SeaLab Adadvanced Information Retrieval*.

Next an analogical circuit able to perform SVM training is developed [12],[13]. Further three algorithmic contributions are presented: a fast approximate solution method for RLS for classification particularly useful in power-constrained devices [14], some improvements on Random Neural Networks [23] and an efficient method to detect non stationarity of given data by using clustering [15],[16]. A last work [24],[25], consists in the application of machine learning methods for classification of magnetic signals for port protection.

4.1 Text Clustering for Security Applications

Text-mining methods have become a key feature for homeland-security technologies, as they can help explore effectively increasing masses of digital documents in the search for relevant information. The automated surveillance of information sources is of strategic importance to effective homeland security [123],[124],[125]. The increased availability of data-intensive heterogeneous sources provides a valuable asset for the intelligence task, but such a situation poses the major issue of sifting security-relevant information from clutter. Data-mining methods have therefore become a key feature for security-related technologies,[125],[126] as they can help explore effectively increasing masses of digital data in the search for relevant information. In fact, the pristine view of using mining methods for pinpointing critical situations is progressively fading away due to the unattainable classification accuracy [127] by contrast, the use of data-mining tools for the discovery and acquisition of strategic information is more and more widely accepted [123],[126],[128].

Text mining techniques provide a powerful tool to deal with the large amounts of text data [129],[130],[131],[132],[133] (both structured and unstructured) that are gathered from any multimedia source (e.g., from Optical Character Recognition, from audio via speech transcription, from web-crawling agents, etc.). The general area of text-mining methods comprises various approaches [133]: detection/tracking tools continuously monitor specific topics over time; document classifiers label individual files and build up models for possible subjects of interest; clustering tools process documents for detecting relevant relations among those subjects. As a result, text mining can profitably support intelligence and security activities in identifying, tracking, extracting, classifying and discovering patterns, so that the outcomes can generate alerts notifications accordingly [134],[135],[136],[137].

This work addresses document clustering and presents a dynamic, adaptive clustering model to arrange unstructured documents into content-based homogeneous groups. Clustering tools can support security applications in the development of predictive models of the observed phenomenon that au-

tomatically derive the thematic categories from the data. Hence, intelligence analysis can exploit the unsupervised nature of document clustering to tackle:

- scenarios that cannot rely on a set of predefined topics or
- scenarios that cannot be properly modeled by a classifier because of the non-stationary nature of the underlying phenomenon, which requires continuous updating for incorporating unseen patterns.

This section presents a model for document clustering that arranges unstructured documents into content-based homogeneous groups. The overall paradigm is hybrid because it combines pattern-recognition grouping algorithms with style-based processing. First, a hybrid metric measures distances between documents, by combining a content-based with a text-style analysis; the metric considers both lexical properties and the structure and styles that characterize the processed documents. Secondly, the model relies on Kernel K-Means for clustering. As a result, the major novelty aspect of the proposed approach is to exploit the implicit mapping of RBF kernel functions to tackle the crucial task of normalizing similarities while embedding text-style information in the whole mechanism. In addition, the present work exploits several real-world benchmarks to compare the performance of the conventional kernel k-means algorithm and the here proposed kernel k-means clustering schemes, which apply Johnson-Lindenstrauss-type random projections for a reduction in dimensionality before clustering. Experimental results show that the document clustering framework based on kernel k-means provide an effective tool to generate consistent structures for information access and retrieval in particular in security related domains.

4.1.1 Document Clustering in Text Mining and Security

Huge quantities of valuable knowledge are embedded in unstructured texts that are gathered from heterogeneous multimedia sources, ranging from hard-copy documents via Optical Character Recognition, to audio via speech transcription, to link analysis mining via web-crawling agents, etc. The resulting mass of data gives law-enforcement and intelligence agencies a valuable

asset, but also poses the major issue of extracting and analyzing structured knowledge from unstructured texts. Text-mining technologies in security applications [128],[134],[135],[136],[137] can automate, improve and speed up the analysis of existing datasets, with the goal of preventing criminal acts by the cataloguing of various threads and pieces of information, which would remain unnoticed when using traditional means of investigation. In general, the text-mining process may involve different sub-goals, such as information extraction, topic tracking, summarization, categorization, clustering, concept linkage, information visualization, and question answering. For prevention, text mining techniques can help identify novel “information trends” revealing new scenarios and threats to be monitored; for investigation, these technologies can help distil relevant information about known scenarios whose actors, situations and relations must be completed, structuring them in patterns really usable for end-users in conjunction with the instruments and methods they daily use.

Within the text mining framework, the present work addresses document clustering, which represents one of the most effective techniques to organize documents in an unsupervised manner. Clustering tools can support the development of predictive models of the observed phenomena, and derive automatically the thematic categories embedded in the data. Therefore, such an unsupervised framework makes document clustering a promising solution for those intelligence tasks that either lack a set of predefined topics, or cannot rely on an exhaustive training set. Clustering can be used to refine and continuously maintain a model of a known distribution of documents, and therefore supports investigation activities by a tracking action. At the same time, clustering can pinpoint emerging, unknown patterns by identifying people, objects, or actions that deserve resource commitment or attention, and thereby support prevention by a novelty-detection capability.

When dealing with text documents, clustering techniques exploits machine learning, natural language processing (NLP), information retrieval (IR), information extraction (IE) and knowledge management to discover new, previously unknown information by automatically extracting information from different written resources. [138] In the following, the document retrieval



Figure 4.1: A generic process model for a document-clustering application

model is outlined, then, the specific document clustering problem is addressed.

4.1.1.1 Document Indexing

A text mining framework should always be supported by an information extraction (IE) model, [139], [140], [103] which is designed to pre-process digital text documents and to organize the information according to a given structure that can be directly interpreted by a machine learning system; in this regard, Fig. 4.1 sketches a generic process model for a document-clustering application. Actually, IE defines how a document and a query are represented and how the relevance of a document to a query is computed.

Document retrieval models typically describe a document D as a set of representative tokens called *index terms*, which result from a series of operations performed on the original text. Stop-words removal and stemming typically are among those operations: the former takes out frequent and semantically non-selective expressions from text; stemming simplifies inflectional forms (sometimes derivationally related forms) of a word down to a common radix form (e.g., by simplifying plurals or verb persons).

Thus, a document D is eventually reduced to a sequence of terms and is represented as a vector, which lies in a space spanned by the *dictionary* (or vocabulary) $T = \{t_j; j = 1, \dots, n_T\}$. The dictionary collects all terms used to represent any document D , and can be assembled empirically by gathering the terms that occurs at least once in a document collection $\mathcal{D} = \{D_1, \dots, D_n\}$. As a consequence, by this representation one loses the original relative ordering of terms within each document. Different models [139], [140], [103] can

be used to retrieve index terms and to generate the vector that represents a document, D . The classical Boolean model involves a binary-valued vector, $\mathbf{v} = \{b_j; j = 1, \dots, n_T\}$, in which each bit component, $b_j \in \{0, 1\}$, just indicates the presence (absence) of term t_j in the document. This approach is most effective for fast searching but proves inaccurate when content-based analysis is required, because it does not render the actual distributions of terms within the considered documents.

The *vector space* model [103] is the most widely used method for text mining and, in particular, for document clustering. Given a collection of documents \mathcal{D} , the vector space model represents each document D as a vector of real-valued weight terms $\mathbf{v} = \{w_j; j = 1, \dots, n_T\}$. Each component of the n_T -dimensional vector is a non-negative *term weight*, w_j , that characterizes the j -th term and denotes the relevance of the term itself within the document D .

Several approaches have been proposed to compute term weights [141]. The popular *tf-idf* weighting scheme [141],[142] relies on two basic assumptions:

- the number of occurrences of a term t in a document D is proportional to the importance of t within D (term frequency, *tf*);
- the number of occurrences of t across different documents is *inversely* proportional to the discriminating power of t , i.e., a term that appears frequently throughout a set of documents is not effective in localizing a specific document in the set (inverse document frequency, *idf*).

Weights computed by *tf-idf* techniques are often normalized so as to contrast the tendency of *tf-idf* to emphasize lengthy documents. Document indexing is a necessary and critical component of any text mining tool, especially because it allows a system to filter out irrelevant information and attain an efficient and cogent representation of content. Such results may be used for document retrieval, as is the case for typical search engines in the security-related applications that are the scope of this framework, however, the index-based representation constitutes an intermediate step for comparing docu-

ments and arranging them into homogeneous groups, which is the ultimate purpose of the clustering engine.

4.1.1.2 Document Clustering

Clustering is conventionally ascribed to the realm of pattern recognition and machine learning [76]. When applied to text mining, clustering algorithms are designed to discover groups in the set of documents such that the documents within a group are more similar to one another than to documents of other groups. As opposed to text categorization [133] in which predefined categories enter the learning procedure, document clustering follows an unsupervised approach to search, retrieve, and organize key topics when a proper set of categories cannot be defined *a-priori*. The unsupervised paradigm can address challenging scenarios, in which local episodes of interest can fade away in the clutter of very large datasets, where events or profiles are ambiguous, unknown, or possibly changing with respect to the original models.

The document clustering problem can be defined as follows. One should first define a set of documents $D = \{D_1, \dots, D_n\}$, a similarity measure (or distance metric), and a partitioning criterion, which is usually implemented by a cost function. *Flat* clustering creates a set of clusters without any *a-priori* assumption about the structure among clusters; it typically requires that the number of clusters to be specified in advance, although adaptive methods exist for determining the cluster cardinality adaptively. Hence, one sets the desired number of clusters, K , and the goal is to compute a membership function f such that minimizes the partitioning cost with respect to the similarities among documents. On the other hand, *hierarchical* clustering arranges groups in a structural, multilevel fashion and does not require a pre-specified number of clusters, instead a branching policy is needed. Hierarchical clustering need not define the cardinality, K , because it applies a series of nested partitioning tasks, which eventually yield a hierarchy of groups. In addition to that choice between a flat or a hierarchical scheme, three main issues should be addressed when designing the overall clustering framework.

The first issue is the curse of dimensionality. When using a vector-space

approach, documents lie in a space whose dimensionality typically ranges from several thousands to tens of thousands. Nonetheless, most documents normally contain a very limited fraction (1%–5%) of the total number of admissible terms in the entire vocabulary, hence the vectors representing documents are very sparse. This can make learning extremely difficult in such a high-dimensional space, especially due to so-called the curse of dimensionality. It is typically desirable to project documents preliminarily into a lower-dimensional subspace, which preserves the semantic structure of the document space but facilitates the use of traditional clustering algorithms. Several methods for low-dimensional document projections have been proposed, [75] such as spectral clustering, clustering using the Latent Semantic Index (LSI), clustering using the Locality Preserving Indexing (LPI), and clustering based on nonnegative matrix factorization [143]. Those methods are quite popular but also exhibit theoretical and practical drawbacks. Both the LSI and the LPI model rely on Singular Value Decomposition (SVD), which optimizes a least-square criterion and best performs when data are characterized by a normal distribution. In fact, the latter assumption does not hold in the general case of term-indexed document matrixes. Besides, LSI, LPI and spectral clustering all require the computation of eigenvalues; as such, these methods often prove both heavy from a computational viewpoint and quite sensitive to outliers. Nonnegative matrix factorization (NMF) differs from other rank reduction methods for vector spaces, especially because of specific constraints that produce nonnegative basis vectors. However, the iterative update method for solving NMF problem is computationally expensive and produces a non-unique factorization.

The second issue in setting up an effective clustering process is the definition of the similarity measure. Since the partitioning criterion often relates strictly to the similarity measure, the choice of the underlying metrics is critical for getting meaningful clusters. For documents, it is normal to address some content-based similarity, and most clustering tools adopt the vector-space model because such a framework easily supports the popular cosine

similarity:

$$\text{sim}(D_i, D_j) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{|\mathbf{v}_i| |\mathbf{v}_j|} \quad (4.1)$$

where \mathbf{v}_i is the vector representation of the document D_i and the operator (\cdot) denotes the conventional inner product in the vector space. The normalization implied by the denominator in (4.1) prevents that two documents having similar distributions of terms appear distant from each other just because one is much longer than the other. In fact, the cosine similarity seems to not outperform the conventional Euclidean distance when high dimensional spaces are concerned [144]

The third issue in clustering for text mining concerns the specific algorithm to be implemented. Although the literature offers a wide variety of clustering algorithms, the majority of research in text mining involves three approaches, namely, *k-means* clustering, Self Organizing Maps (SOMs), and the Expectation-Maximization (EM) algorithm. Alternative approaches include models based on fuzzy clustering techniques [145],[146]. Furthermore, on a slightly different perspective, the works of Hammouda et al. [147] and Chim et al. [148] proposed document-clustering schemes exploiting a phrase-based document similarity. The former scheme [147] exploits the Document Index Graph (DIG), which indexes the documents while maintaining the sentence structure in the original documents; the latter scheme [148] is based on the Suffix Tree Document (STD) model [149].

4.1.2 Hybrid Approach to Kernel K-Means Clustering

The hybrid approach described in this Section combines the specific advantages of content-driven processing with the effectiveness of an established pattern-recognition grouping algorithm. The ultimate goal of the clustering process is to group a set of documents into a (possibly adaptive) structure of clusters, which contain documents that are affine for both contents and structural features. The presentation of the method will mostly address the three main issues discussed in the previous section.

With respect to the dimensionality problem, the method extends a conventional vector-space representation, mostly because that approach was shown to suffer from curse-of-dimensionality problems even in the presence of sparse matrixes. Even though a document is represented by the set of composing terms annotated with frequency and positional information, the subsequent processing steps do not involve any matrix-intensive computation such as SVD-related methods. This is intentionally done to reduce sensitivity to outliers; moreover, the dimensionality problem is strictly related to the effectiveness of the actual clustering strategy adopted, and the proposed approach facilitates kernel-based implementations as will be clarified in the following.

Document similarity is defined by a content-based distance, which combines a classical distribution-based measure with a behavioural analysis of the style features of the compared documents. The involved metric thus considers lexical properties, the structures, and some style features that characterize the processed documents. The method intentionally does not consider semantic-based information (such as the use of ontologies and deductive methods

In the following, $\mathcal{D} = \{D_u; u = 1, \dots, D_n\}$ will denote the *corpus*, holding the collection of documents to be clustered. The set $T = \{t_j; j = 1, \dots, n_T\}$ will denote the *vocabulary*, which is the collection of terms that occur at least one time in D after the pre-processing steps of each document $D \in \mathcal{D}$ (e.g., stop-words removal, stemming). Accordingly, \mathbf{d} will represent the document D as a sequence of indexes; thus $d_u = \{d_q^{(u)}; q = 1, \dots, n_E^{(u)}\}$, where $d_q^{(u)}$ is the index in T of the q -th term in D_u and $n_E^{(u)}$ is the number of terms in document D_u . Obviously, the order of the indexes in \mathbf{d}_u matches the relative ordering of the terms in the document.

4.1.2.1 Document distance measure

A novel aspect of the method described here is the use of a document-distance that takes into account both a conventional content-based similarity metric and a behavioral similarity criterion. The latter term aims to improve the

overall performance of the clustering framework by including the structure and style of the documents in the process of similarity evaluation. To support the proposed document distance measure, a document D is here represented by a pair of vectors, \mathbf{v}' and \mathbf{v}'' . Vector \mathbf{v}' actually addresses the content description of a document D ; it can be viewed as the conventional n_T -dimensional vector that associates each term $t \in T$ with the normalized frequency, tf , of that term in the document D . Therefore, the k -th element of the vector \mathbf{v}' is defined as:

$$v'_{k,u} = \frac{tf_{k,u}}{\sum_{l=1}^{n_T} tf_{l,u}} \quad (4.2)$$

where $tf_{k,u}$ is the frequency of the k -th term in document D_u . Thus \mathbf{v}' represents a document by a classical vector model, and uses term frequencies to set the weights associated to each element. Thanks to its local descriptive nature, \mathbf{v}' can be worked out without using global properties of the corpus D such as the idf measure.

From a different perspective, the structural properties of a document, D , are represented by a set of probability distributions associated with the terms in the vocabulary. Each term $t \in T$ that occurs in D_u is associated with a distribution function that gives the spatial probability density function (pdf) of t in D_u . Such a distribution, $p_{t,u}(s)$, is generated under the hypothesis that, when detecting the k -th occurrence of a term t at the normalized position $s_k \in [0, 1]$ in the text, the spatial pdf of the term can be approximated by a Gaussian distribution centered around s_k . In other words, the proposed behavioral similarity criterion supposes that if the term t_j is found at position s_k within a document, another document with a similar structure is expected to include the same term at the same position or in a neighborhood thereof, with a probability defined by a Gaussian pdf. Although empirical, the (practically reasonable) assumption is that the spatial probability density function of a term t in a document can characterize the document itself. In fact, one must be aware that such an assumption may not hold for heavily unstructured text data. In this respect, the eventual distance value, $\Delta(D_u, D_v)$ between two documents should properly mix the relative contribution of the

two similarity criteria according to the applicative scenario; such aspect will be addressed later in this section. Actually, the ultimate validation of the document-distance measure will only stem from testing the empirical performance of the clustering framework on the experimental domain.

To derive a formal expression of the pdf, assume that the u -th document, D_u , holds n_o occurrences of terms after simplifications; if a term occurs more than once, each occurrence is counted individually when computing n_o , which can be viewed as a measure of the length of the document. The spatial pdf can be defined as:

$$p_{t,u}(s) = \frac{1}{A} \sum_{k=1}^{n_o} G(s_k, \lambda) = \frac{1}{A} \sum_{k=1}^{n_o} \frac{1}{\sqrt{2\pi}\lambda} \exp \left[-\frac{(s - s_k)^2}{\lambda^2} \right] \quad (4.3)$$

where A is a normalization term.

In practice, one uses a discrete approximation of (4.3) first by segmenting evenly the document D into S sections. Then, an S -dimensional vector is generated for each term $t \in T$, and each element estimates the probability that the term t occurs in the corresponding section of the document.

As a result, $\mathbf{v}''(D)$ is an array of n_T vectors having dimension S . Figures 4.2,4.3 sketches the procedure that supports the computation of \mathbf{v}'' for a term t_j . First, the document is evenly segmented into S parts; then, the term frequency for each segment $d_i^{(s)}$ is worked out. Finally, the vector \mathbf{v}'' is built up as a superposition of Gaussian distributions. The behavioral component, ascribed to term t , of the style-related distance between a pair of documents can therefore be computed as the distance between the two corresponding pdf's:

$$\Delta_t^{(b)}(D_u, D_v) = \int_Z [p_{t,u}(z) - p_{t,v}(z)]^2 dz \quad (4.4)$$

Vector \mathbf{v}' and vector \mathbf{v}'' support the computations of the frequency-based distance, $\Delta^{(f)}$ and the behavioral distance, $\Delta^{(b)}$, respectively. The former term is usually measured according to a standard Minkowski distance, hence

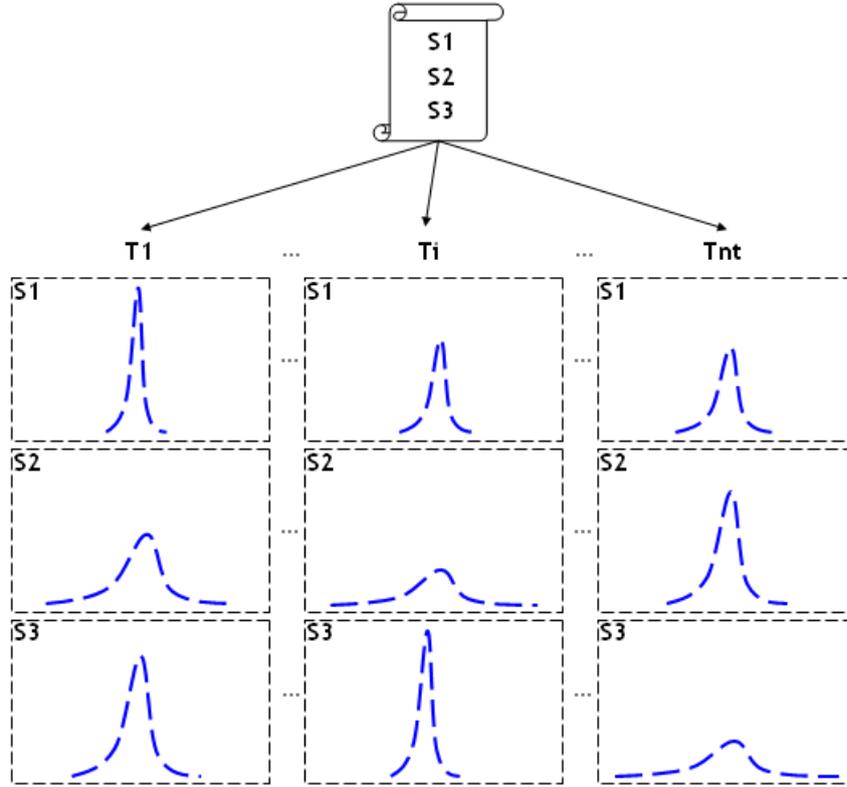


Figure 4.2: A document partitioned in 3 sections and terms. $T = t_j; j = 1, \dots, n_T$ Gaussian densities in each section

the content distance between a pair of documents (D_u, D_v) is defined by:

$$\Delta^{(f)}(D_u, D_v) = \left[\sum_{k=1}^{n_T} |v'_{k,u} - v'_{k,v}|^p \right]^{1/p} \quad (4.5)$$

The present approach adopts the value $p = 1$ and therefore actually implements a Manhattan distance metric. The term computing behavioral distance, $\Delta^{(b)}$, applies an Euclidean metric to compute the distance between probability vectors \mathbf{v} . Thus:

$$\Delta^{(b)}(D_u, D_v) = \sum_{k=1}^{n_T} \Delta_{t_k}^{(b)}(D_u, D_v) \quad (4.6)$$

Both terms (4.5) and (4.6) contribute to the computation of the eventual distance value, $\Delta(D_u, D_v)$, which is defined as follows:

$$\Delta(D_u, D_v) = \alpha \Delta^{(f)}(D_u, D_v) + (1 - \alpha) \Delta^{(b)}(D_u, D_v) \quad (4.7)$$

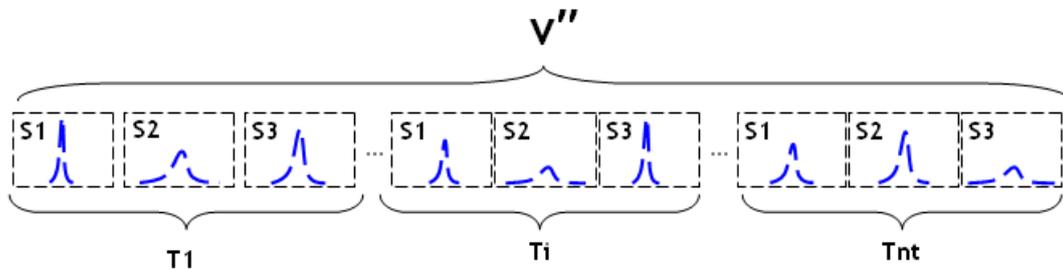


Figure 4.3: A document partitioned in 3 sections, terms $T = t_j; j = 1, \dots, n_T$ and vector v'' representation Gaussian densities in each section

where the mixing coefficient $\alpha \in [0, 1]$ weights the relative contribution of $\Delta^{(f)}$ and $\Delta^{(b)}$. It is worth noting that the distance expression (4.7) obeys the basic properties of non-negative values and symmetry that characterize general metrics, but does not necessarily satisfy the triangular property.

4.1.3 The document-clustering framework

In the following, a comprehensive overview of the eventual document-clustering framework will be presented. Furthermore, the issue of curse of dimensionality will be addressed by proposing a dimensionality reduction model based on Random Projections. Finally, the computational complexity of the proposed framework will be analyzed.

4.1.3.1 The document-clustering algorithm

The proposed framework for document clustering combines the previously described metric with pattern-recognition grouping algorithms. The following pseudocode outlines the complete framework:

The framework can support both flat clustering and hierarchical clustering. When flat clustering is addressed, the algorithm partitions the set of documents D into K clusters, where K is an input parameter; however, no explicit structure is produced that would relate clusters to each other. Conversely, the hierarchical clustering model applies a top-down paradigm to generate a cluster hierarchy. Hence, the procedure starts with all the documents in single cluster. Then, the cluster is split using the flat clustering tech-

Algorithm 7 The document-clustering procedure**Require:** A corpus of n documents, \mathcal{D} **Ensure:** Clusters membership \mathbf{m}

- 1: $\forall D_i \in \mathcal{D}$ gets \bar{D}_i as follows:
 1. Apply stop-words removal
 2. Apply stemming
- 2: Build the vocabulary $T = \{t_j; j = 1, \dots, n_T\}$ as the set of terms that occurs at least one time in the corpus.
- 3: $\forall D_i \in \mathcal{D}$
 1. build $\mathbf{v}'_{\bar{D}_i}$ as per (4.2)
 2. build $\mathbf{v}''_{\bar{D}_i}$ as per (4.3)
- 4: Build the rbf dot matrix \mathbf{K} of elements $K(\bar{D}_i, \bar{D}_j) = e^{-[\Delta_{ij}]^2}$
- 5: Run Kernel K-Means on \mathbf{K} and return the membership \mathbf{m}

nique; this procedure is applied recursively until some stopping criterion is met. In the present work, only flat clustering is used for experiments.

4.1.3.2 Dimension reduction for document clustering by using random projections

As anticipated the curse of dimensionality represents a crucial issue when dealing with document clustering. By adopting the vector space model, the total number of unique terms in a text data set represents the number of dimensions, which is usually in the thousands. Nonetheless, sparsity is an accompanying phenomenon of such data representation model. Therefore, in the recent years several works have addressed the problem of dimensionality reduction for document clustering tools [150], [151], [152], [153], [154], [155]. The present research tackles this significant aspect by comparing the performance of the conventional kernel k-means algorithm with that of a kernel k-means clustering scheme recently proposed by Biau et al. [156], which exploits random projections for a reduction in dimensionality before clustering. In that work, a notable theoretical discussion led to the conclusion

that Johnson-Lindenstrauss-type random projections to lower-dimensional subspaces are particularly well suited for clustering purposes, as they can outperform other dimension reduction schemes based on factorial methods (e.g. Principal Component Analysis). Over the years, the probabilistic method of Random Projections has allowed for the original proof of Johnson and Lindenstrauss to be greatly simplified, while at the same time giving conceptually simple randomized algorithms for constructing the embedding [80] (see Chapter 1 for brief introduction to Random Projections). Obviously, the choice of the random matrix \mathbf{R} is one of the key points of interest. Although the elements r_{ij} of \mathbf{R} are often Gaussian distributed, the present research exploited the theoretical results proposed by Achlioptas,[81] which showed that there are simpler ways of producing Random Projections. Thus, \mathbf{R} can be generated as follows:

$$r_{ij} = \sqrt{3} \cdot \begin{cases} +1 & \text{with probability } 1/6 \\ 0 & \text{with probability } 2/3 \\ -1 & \text{with probability } 1/6 \end{cases} \quad (4.8)$$

In this work, random projections for reduction in dimensionality are applied to the term-by-document matrix that results from working out vector $\mathbf{v}'(\bar{D}_i)$ for every $D_i \in \mathcal{D}$. Therefore, \mathbf{X} is a set of n documents lying in a n_T dimensional space and \mathbf{R} is a $n_T \times n_r$ matrix computed as indicated above where n_r is the number of desired dimensions after the projection.

The following section presents experimental results with and without random projections on different real world text mining domains: Enron database [157], Reuters dataset [100] and Newsgroup 20 collections [100].

4.1.4 Experimental Results

Three real-world benchmarks provided the experimental domains for the proposed document-clustering framework: the Reuters database [100], the well known Newsgroup 20 emails collection [100], and the largest real email corpus in the public domain, the Enron mail dataset [157]. The former and the second were chosen to assess the method performance on a public source

of information, which represents a typical framework of today's trends in data gathering and analysis for security and intelligence. The latter dataset made it possible to validate the operation of the clustering principle in a set of complex scenarios involving non-trivial security issues.

The Reuters-21578 corpus includes 21,578 documents, which appeared on the Reuters newswire in 1987. One or more topics have been manually added to each document. Actually, the whole database involves 135 different topics derived from economic subject categories; indeed, only 57 topics have at least twenty occurrences. The Reuters-21578 corpus represents a standard benchmark for content-based document management. This work exploits such corpus to show that the document clustering framework based on kernel k-means provide an effective tool to generate consistent structures for information access and retrieval.

The 20 Newsgroups corpus includes 20,000 messages collected by using as source 20 different newsgroups. Accordingly, 20 topics have been used to categorize the documents in the corpus. In this case, the documents are almost evenly distributed over the different topics. The 20 Newsgroups database provided the second experimental domain for the proposed framework. The experiments involved two different corpora, \mathcal{D}_{N_1} and \mathcal{D}_{N_2} , worked out from such database. Corpus \mathcal{D}_{N_1} and corpus \mathcal{D}_{N_2} , were generated by using the criteria proposed in the work by Jing et al [158]. Thus, \mathcal{D}_{N_1} included all the documents (3894 elements) associated to the categories: *comp.graphics*, *rec.sport.baseball*, *sci.space*, and *talk.politics.mideast*; \mathcal{D}_{N_2} included all the documents (3929 elements) associated to the categories: *comp.graphics*, *comp.os.ms-windows*, *rec.autos*, and *sci.electronics*.

On the other hand, the Enron mail dataset provides a reference corpus to test text-mining techniques that address intelligence applications. The Enron mail corpus was posted originally on Internet by the *Federal Energy Regulatory Commission* (FERC) during its investigation on the Enron case. FERC collected a total of 619,449 emails from 158 Enron employees, mainly senior managers. Each message included: the email addresses of the sender and receiver, date, time, subject, body and text. The original set suffered from document integrity problems, hence an updated version was later set up

by SRI International for the CALO project. Eventually, William Cohen from Carnegie Mellon University put the cleaned dataset TM22 for researchers in March 2004. Other processed versions of the Enron corpus have been made available on the web, but were not considered in the present work because the CMU version made it possible fair comparison of the obtained results with respect to established, reference corpora in the literature.

The set of messages covered a widest range of topics, originating from a vast community of people who did not form a closed set. A few people wished to conceal both the extent of their connections and the contents of their discussions; at the same time, by far the large majority of messages were completely innocent. As a result, the Enron data set provided a good experimental domain to evaluate the ability of a text mining framework in a ‘needle-in-a-haystack’ scenario, which closely resembled a typical situation in security-related applications such as counterterrorism.

4.1.4.1 Reuters-21578

The experimental session based on the Reuters-21578 database involved a corpus \mathcal{D}_R including 8,267 documents out of the 21,578 originally provided by the database. The eventual corpus \mathcal{D}_R was obtained by adopting the criterion already used in the work of Cai et al. [154]. First, all the documents with multiple topics were discarded. Then, only the documents associated to topics having at least 18 occurrences have been included in \mathcal{D}_R . As a result, the corpus featured 32 topics. The clustering performance of the proposed methodology was evaluated by analyzing the results obtained in three experiments. In the experiments, a flat clustering paradigm partitioned the documents in \mathcal{D}_R , by using three different settings of the metric weight parameter, $\alpha = \{0.3, 0.5, 0.7\}$, as per (4.7). When adopting $\alpha = 0.3$ or $\alpha = 0.7$ in (4.7), either component $\Delta^{(f)}$ or $\Delta^{(b)}$, was predominant; in the experiment featuring $\alpha = 0.5$, the quantities $\Delta^{(f)}$ and $\Delta^{(b)}$ evenly contributed to the measured distance between each pair of documents.

Table 4.1 outlines the results obtained with the setting $\alpha = 0.3$. The performances of the proposed clustering framework were evaluated by using the *purity* parameter. Let n_j denote the number of elements lying in a cluster C_j

K	pur_{ov}	$pur_l(k)$	$pur_h(k)$	ψ
20	0.712108	0.252049	1	109
40	0.77138	0.236264	1	59
60	0.81154	0.175	1	13
80	0.799685	0.181818	1	2
100	0.82666	0.153846	1	1

Table 4.1: Clustering performances obtained on Reuters-21578 with $\alpha = 0.3$

and let n_{mj} be the number of elements of the class m in the cluster C_j . Then, the purity $pur(j)$ of the cluster C_j is defined as follows:

$$pur(j) = \frac{1}{n_j} \max_m(n_{mj}) \quad (4.9)$$

The overall purity of the clustering results is defined as follows:

$$pur_{ov} = \sum_j \frac{n_j}{n} \cdot pur(j) \quad (4.10)$$

where n is the total number of elements.

The purity parameter (a.k.a classification accuracy) was preferred to other measures of performance (e.g. the F-measures) because it is widely accepted in machine learning classification problems [138].

The evaluations were conducted with different number of clusters K , ranging from 20 to 100. For each experiment, four quality parameters are presented:

1. the overall purity, pur_{ov} , of the clustering result;
2. the lowest purity value $pur_l(k)$ over the K clusters;
3. the highest purity value $pur_h(k)$ over the K clusters;
4. the number ψ of elements (i.e. documents) associated to the smallest cluster.

K	pur_{ov}	$pur_l(k)$	$pur_h(k)$	ψ
20	0.696383	0.148148	1	59
40	0.782267	0.222467	1	4
60	0.809121	0.181818	1	1
80	0.817467	0.158333	1	1
100	0.817467	0.139241	1	2

Table 4.2: Clustering performances obtained on Reuters-21578 with $\alpha = 0.5$

K	pur_{ov}	$pur_l(k)$	$pur_h(k)$	ψ
20	0.690577	0.145719	1	13
40	0.742833	0.172638	1	6
60	0.798718	0.18	1	5
80	0.809483	0.189655	1	2
100	0.802589	0.141732	1	4

Table 4.3: Clustering performances obtained on Reuters-21578 with $\alpha = 0.7$

Tables 4.2 and 4.3 reports the results obtained with $\alpha = 0.5$ and $\alpha = 0.7$, respectively. As expected, experimental results showed that overall purity increased when the number of clusters, K , increased. The value of the overall purity seems to indicate that clustering performances improved when setting $\alpha = 0.3$. Thus empirical outcomes confirmed the effectiveness of the proposed document distance measure, combining the conventional content-based similarity with a behavioral similarity criterion.

A detailed analysis of the clustering performances attained by the proposed framework can be drawn from the graphs presented in 4.4. The graphs report the cumulative distribution of the cluster purity for the different experiments listed in 4.1 (only the experiment with $K=20$ is not included). The reported results showed that, in every experiment, 70% of the obtained clusters exhibited a purity greater than 0.6; furthermore, 50% of the clusters scored a purity greater than 0.9. The reliability of the present document-clustering scheme was indeed confirmed when counting the documents lying in clus-

K	pur_{ov}	$pur_l(k)$	$pur_h(k)$	ψ
20	0.70231	0.126386	1	32
40	0.709084	0.117647	1	2
60	0.731221	0.182109	1	1
80	0.726019	0.121145	1	1
100	0.692029	0.158004	1	1

Table 4.4: Clustering performances obtained on Reuters-21578 with $\alpha = 0.3$ and dimension reduction $n_r = 500$

ters having a purity of 1.0. With an overall number of clusters $K = 40$, a set of 2,575 documents (i.e., 31% of the total number of documents) were assigned to those clusters; when using $K = 100$, that percentage increased to 35% (2,950 documents).

A final set of experiments involved the clustering scheme exploiting random projections for a reduction in dimensionality. This analysis addressed the critical aspect concerning the trade off between clustering accuracy and computational complexity. Such a problem is common in all text-mining applications and can prove especially relevant when using the k-means clustering algorithm. The performance of this clustering scheme was evaluated on the corpus \mathcal{D}_R by setting $\alpha = 0.3$; the dimension of the eventual reduced space was set to $n_r = 500$ and $n_r = 100$. Table 4.4 and Table 4.5 report the results obtained with these set up by varying the desired number of clusters K .

Unsurprisingly, the clustering performances obtained by exploiting dimension reduction were slightly inferior to those attained by the conventional clustering scheme (Table 4.1). Indeed, one should take into account that by setting $n_r = 500$ the original term-by-document matrix had been reduced of a factor 100. Nonetheless, Table 4.4 and Table 4.5 show that the overall purity attained by the model was satisfactory even in the presence of a substantial reduction in dimensionality.

These results can be compared with those obtained on the same dataset by the clustering framework proposed by Cai et al.,[154] which introduced

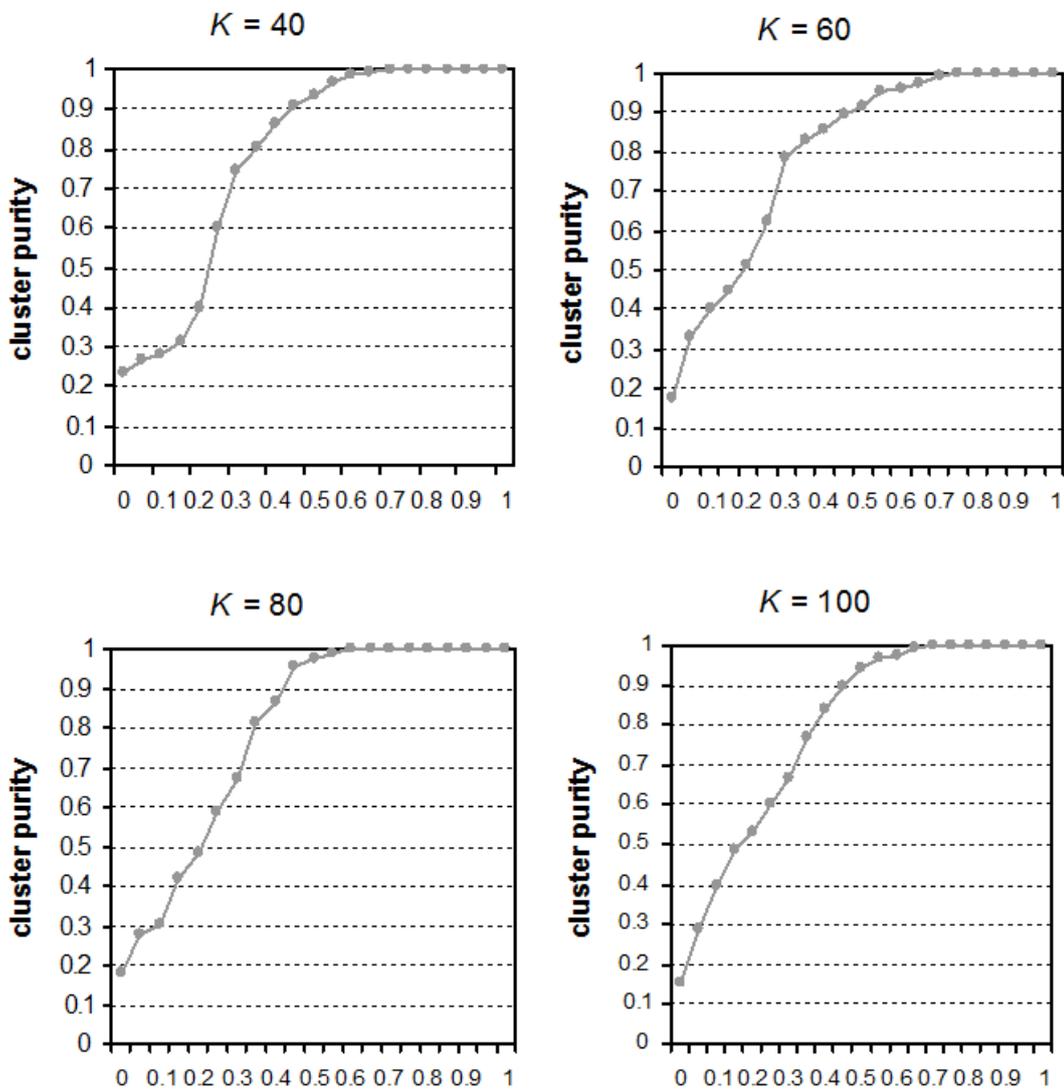


Figure 4.4: The cumulative distributions of the cluster purity for the experiments reported in Table 4.1

K	pur_{ov}	$pur_l(k)$	$pur_h(k)$	ψ
20	0.666626	0.106776	1	31
40	0.69723	0.115813	1	2
60	0.705213	0.149718	1	1
80	0.721785	0.11674	1	1
100	0.739204	0.137662	1	1

Table 4.5: Clustering performances obtained on Reuters-21578 with $\alpha = 0.3$ and dimension reduction $n_r = 100$

a clustering scheme combining the conventional k-means algorithm with a dimensionality reduction model based on the Locality Preserving Index (LPI). The results presented in that work showed that the LPI-based clustering scheme, although computationally demanding, can outperform other well-known methodologies, such as LSI-based clustering and spectral clustering algorithms.

When applied to the Reuters database, the LPI-based clustering scheme attained an average purity of 0.77; in that work, evaluations were conducted with different number of clusters, ranging from two to ten. Thus, the clustering performances of the LPI-based clustering are slightly better than those obtained by the proposed clustering scheme, which exploits random projections for dimensionality reduction. However, as anticipated the scheme based on random projections outperforms the LPI reduction model in terms of computational complexity: while dimensionality reduction by using random projections is supported by a straightforward matrix multiplication, the LPI scheme is actually based on the computationally expensive Singular Valued Decomposition (SVD). Therefore, the dimension reduction model based on random projections proposed in the present work seems to guarantee a satisfactory tradeoff between performance and complexity.

The effectiveness of the proposed document-clustering framework was eventually confirmed by a comparison with the document-clustering model using nonnegative matrix factorization (NMF) introduced by Shahnaz et al. [155] When applied to the Reuters database, the NMF-based model attained

K	pur_{ov}	$pur_l(k)$	$pur_h(k)$	ψ
20	0.755778	0.433409	1	58
40	0.773754	0.314286	1	29
60	0.792244	0.33871	1	9
80	0.819723	0.290323	1	4
100	0.811505	0.311475	1	1

Table 4.6: Clustering performances obtained on \mathcal{D}_{N1} with $\alpha = 0.3$

an overall purity superior to 0.7 only in experimental sessions involving 6 topics (or less) out of the 135 originally included in the database. Therefore, the prediction accuracy obtained by the model introduced here outperformed the one attained by the NMF-based model.

4.1.4.2 Newsgroup 20

Table 4.6 and Table 4.7 present the results obtained with \mathcal{D}_{N1} and \mathcal{D}_{N2} , respectively with no random projections. In both cases, the setting $\alpha = 0.3$ has been used. Numerical figures show that the system attained with \mathcal{D}_{N1} an overall purity always superior to 0.75; however, clustering performances slightly worsen with corpus \mathcal{D}_{N2} . Such a result can be explained by analyzing the characteristics of the two corpora. Corpus \mathcal{D}_{N1} involves categories semantically well separated, while corpus \mathcal{D}_{N2} . When comparing these results with those obtained on the same testbed in the work by Jing et al [158] one should take into account the differences in the set up of the two experiments. In that research, a modified version of the conventional k-means algorithm, the Entropy Weighting k-Means Algorithm, was used for document clustering. The experiments actually involved a sub-sampled version of the two corpora \mathcal{D}_{N1} and \mathcal{D}_{N2} ; hence, two corpora including 400 documents each were eventually used to test the proposed clustering scheme, which attained an overall purity larger than 0.88 in both experiments.

K	pur_{ov}	$pur_l(k)$	$pur_h(k)$	ψ
20	0.627895	0.303704	1	20
40	0.679817	0.298701	1	14
60	0.691016	0.295775	1	4
80	0.657928	0.265306	1	5
100	0.695597	0.349515	1	5

Table 4.7: Clustering performances obtained on \mathcal{D}_{N2} with $\alpha = 0.3$

4.1.4.3 Enron Dataset

The experimental session based on the Enron mail corpus involved two different experiments. The first experiment exploited the dataset made available on the web by Ron Bekkerman from University of Massachusetts Amherst. The dataset [157] collects email from the directories of seven former Enron employees: *beck-s*, *farmer-d*, *kaminski-v*, *kitchen-l*, *lokay-m*, *sanders-r* and *williams-w3*. Each of these users had several thousand messages, with *beck-s* having more than one hundred folders. The goal of the first experiment was to evaluate the ability of the proposed clustering framework to extract key elements from a heterogeneous scenario, in which one cannot rely on a set of predefined topics. All the seven folders were processed to remove non-topical folders such as *all_documents*, *calendar*, *contacts*, *deleted_items*, *discussion_threads*, *inbox*, *notes_inbox*, *sent*, *sent_items* and *sent_mail*.

For the purpose of increasing complexity, Bekkerman’s dataset was augmented by including the email folder of one of the former executives of Enron, Vice President Sara Shackleton. The underlying hypothesis was that email contents might also be characterized by the role the mailbox owner played within the company. Toward that end, when applying the clustering algorithm, only the ‘body’ sections of the emails were used, and sender/receiver, date/time info were discarded.

The experiment involved 24,355 emails. Table 4.8 reports on the results obtained by this experiment and shows the terms that characterize each of the clusters provided by the clustering framework. For each cluster, the most

j	$ C_j $	Most frequent words
1	1825	ect, Kaminski, research, placeCityHouston, manag, energy
2	8416	deal, hpl, gas
3	1881	Kitchen, deal, gas
4	1210	dbcaps97data, database, iso, error
5	1033	epmi, mmbtu, placeStateCalifornia, gas, northwest
6	2094	ect, manag, risk, trade, market,
7	1239	deal, work, meet
8	1522	message, email, http, subject,
9	1091	market, energy, trade, price, share, stock, gas, dynegi
10	4044	ect, Shackleton, agreement, trade, isda

Table 4.8: Results of the first experiment on the Enron dataset: j is cluster index and $|C_j|$ is the cluster size

descriptive words between the twenty most frequent words of the cluster are listed; reported terms actually included peculiar abbreviations: “ect” stands for Enron Capital & Trade Resources, “hpl” stands for Houston Pipeline Company, “epmi” stands for Enron Power Marketing Inc, “mmbtu” stands for Million British Thermal Units, “dynegi” stands for Dynegy Inc, a large owner and operator of power plants and a player in the natural gas liquids and coal business, which in 2001 made an unsuccessful takeover bid for Enron. The results reported in Table 4.8 showed that the document clustering attained some significant outcomes. First, cluster no.10 grouped a considerable portion of the emails ascribed to the Shackleton subset: most frequent words indeed confirm that the word “Shackleton” appears several time in the email bodies, as well as the term “isda,” which stands for International Swaps and Derivatives Association. It is worth noting that the term “isda” never appeared in the list of the most frequent terms of the other nine clusters. Another significant outcome concerned clusters no. 7 and no. 8, which seemed to group all emails that did not deal with technical topics. At the same time, both clusters no. 1 and cluster no. 3 related to Enron employees, Kaminski and Kitchen, respectively. When analyzing cluster no. 4, it turned out that it gathered the email

notifications automatically sent by a database server, which were collected in a subfolder in the mailbox of *williams-w3*.

The second experiment aimed at estimating the ability of the proposed framework to group messages by the same author when considering body text only. This experiment ultimately aimed at verifying the application of text mining technologies in an intelligence-analysis environment, in which fake identities may be bypassed by associating messages to the original authors. Thus the clustering algorithm was tested on a dataset collecting all the emails included in the folder “sent” of six Enron employees randomly selected: *symes-k*, *namec-g*, *lenhart-m*, *delainey-d*, *rogers-b*. Eventually, the corpus included 6,618 emails body; obviously, all information concerning the email addresses of senders/receivers was discarded.

The graph in Figure 4.5 shows the results obtained in this experiment; the *x*-axis gives the number of clusters, whereas the *y*-axis reports on the classification error (an error was detected when ascribing a message to the wrong author). The obtained figures proved the effectiveness of the proposed clustering methodology, which scored a classification error of 15% (i.e. 1,043 emails) when using 60 clusters for the partitioning. Moreover, the analysis of the eventual clusters led to interesting outcomes. In the 60-clusters partitioning, fifteen clusters were assigned, after the calibration procedure, to the author *lenhart-m*. Those clusters actually shared also a great part of the terms included in their own list the most frequent words; but, surprisingly enough, those terms were *weekend*, *tonight*, *party*, *golf*, *gift*, *ticket*, *happy*, *softball*, *hotmail*, *jpg*, *msn*, *love*, *game*, *birthday*, *celebrate adult*, *drink*, *pool*. Hence, the unsupervised clustering procedure revealed that a significant portion of the emails included in the “Sent” folder of the author *lenhart-m* did not deal with themes related to the working activity. Indeed, such outstanding result can be double checked by actually analyzing the emails provided in the Enron database.

4.1.5 Conclusions

Text mining provides a valuable tool to deal with large amounts of unstructured text data. Indeed, in security applications text-mining technologies can

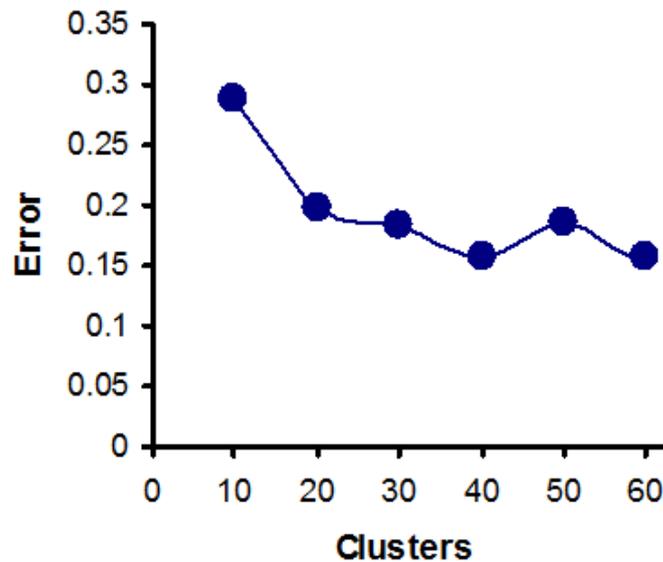


Figure 4.5: Results of the second experiment on the Enron database

automate, improve and speed up the analysis of existing datasets, with the goal of preventing criminal acts by the cataloguing of various threads and pieces of information, which would remain unnoticed when using traditional means of investigation.

Within the text mining environment, document clustering represents one of the most effective techniques to organize documents in an unsupervised manner. Nonetheless, the design of a document-clustering framework requires one to address other crucial aspects in addition to the choice of the specific clustering algorithm to be implemented. A major characteristic of the representation paradigm of text documents is the high dimensionality of the feature space, which imposes a big challenge to the performance of clustering algorithms. Furthermore, the definition of the underlying distance measure between documents is critical for getting meaningful clusters. The hybrid document-clustering approach proposed in this work mostly addresses such issues by combining the specific advantages of content-driven processing with the effectiveness of an established pattern-recognition grouping algorithm. Two crucial novelty aspects characterize the proposed approach. First, distances between documents are worked out by a style-based hyper-

metric. The specific approach integrates a content-based with a user-behavioral analysis, as it takes into account both lexical and style-related features of the documents at hand.

Secondly, the core clustering strategy exploits a kernel-based version of the conventional k-means algorithm to group similar documents; hence, the model exploits the implicit normalization of RBF kernel functions while embedding style information in the whole mechanism. The present research tackled indeed the problem of curse of dimensionality by considering an advanced approach in the vector-space paradigm, which applied Johnson-Lindenstrauss-type random projections for a reduction in dimensionality before clustering. The analysis focused in particular on the critical aspect concerning the trade off between clustering accuracy and computational complexity, which is a general issue in all text-mining applications, and can be particularly relevant when using k-means for document clustering. Experimental results indeed confirmed the consistency of the proposed framework. The hybrid document-clustering approach proved effective when dealing with a standard benchmark for content-based document management. Furthermore, it attained remarkable performances with an experimental domain (Enron) resembling the kind of data collected as part of counterterrorism efforts.

4.2 SVM Analog Circuit Based Learning

Several approaches have been proposed to the effective support of the training process of Support Vector Machines [3]. When learning speed is of paramount importance, one might envision a hardware-based approach to the training strategy, and one should first choose between a digital and an analog approach to the circuit-support strategy.

The aim of this work is to characterize and empirically investigate the aspects and the potentialities of a general circuit model based on the co-content minimization [159].

4.2.1 Hardware SVM

The SVM training formulation implies a constrained quadratic programming problem (CQP) on a convex cost function. The major practical advantage of this property is that polynomial-complexity Quadratic Programming (QP) algorithms ensure convergence to the global minimum.

When the number of free parameters becomes huge (e.g. $n > 10^4$ patterns), several decomposition algorithms have been proposed to cope with the optimization problem of SVM. In such cases, it seems interesting to envision a hardware solution to this problem, especially when considering computational efficiency.

An approach to map SVM learning stage on an analog circuit was proposed in [160], and was characterized by a direct mapping of the SVM learning process into Chua's circuit [159]. The main drawback of that method was the circuit complexity in mapping the linear constraint in (2.97), which was replaced by two inequalities:

$$\begin{aligned} \sum_{i=1}^n y_i \alpha_i &\geq 0 \\ \sum_{i=1}^n y_i \alpha_i &\leq 0 \end{aligned} \quad (4.11)$$

Such a formulation is formally correct but might bring about some issues in reaching a circuit stable state.

The present work shows that by using 2.103 that is an SVM with no bias term, co-content networks can apply effectively to the circuit-supported optimization goal.

4.2.2 Co-Content Minimization Circuits

The basic principle underlying the co-content approach is that the minimum of a given functional can be found by means of a proper analog circuit. Let \mathfrak{R} be a voltage-controlled, reciprocal, multiterminal resistor; for such component (either linear or nonlinear) one can formalize the so-called co-content potential function, $G(\mathbf{v})$, as

$$\Gamma(\mathbf{v}) = \int^v i(\xi) \cdot d\xi \quad (4.12)$$

where \mathbf{v} , \mathbf{i} denote the vectors of voltages and currents respectively, taken as shown in 4.6.

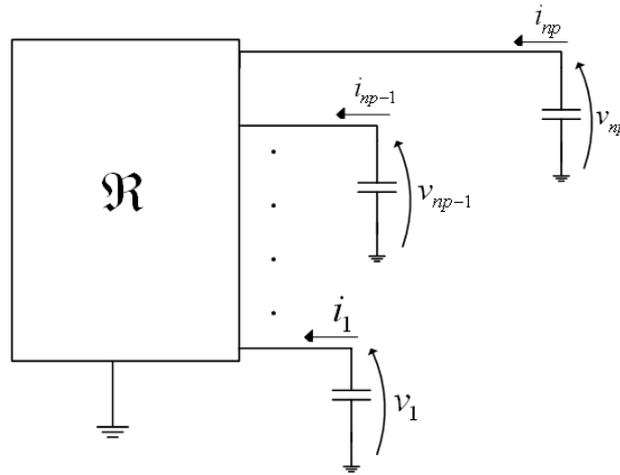


Figure 4.6: Multi-terminal resistive network connected to capacitors

The current vector, \mathbf{i} , can be written as the gradient of the co-content function:

$$i(v) = \nabla \Gamma(v) \quad (4.13)$$

In the circuit obtained by connecting \mathfrak{R} to a set of linear capacitors 4.6, the voltages in \mathbf{v} encode the state variables of the circuit. If one denotes as \mathbf{C} the diagonal matrix containing the (possibly different) capacitance values, theory shows [161] that the time progression of $G(t)$ is ruled by the equation:

$$\frac{d\Gamma}{dt} = (\nabla \Gamma(v))^t \frac{\partial v}{\partial t} = - \left(\frac{\partial v}{\partial t} \right)^t C \frac{\partial v}{\partial t} \leq 0 \quad (4.14)$$

Expression (4.14) points out that, for any initial condition $\mathbf{v}(\mathbf{0})$, $G(t)$ points toward a minimum and that the corresponding stationary value of the current vector in the circuit is $\mathbf{i} \equiv \mathbf{0}$.

This property suggests that one could minimize an arbitrary functional, \mathfrak{S} , by mapping it on the co-content, $G(\mathbf{v})$, of a proper resistive multi-terminal component, connected to a set of linear capacitors as per 4.6. The main structural idea is to bypass digital-based computations in the functional minimization, by directly using the intrinsic computational capabilities offered by the physical laws of circuits [159]. Since the solution is reached in real time as soon as the kernel matrix is computed, the time needed for SVM training reduces considerably respect software implementations; one should also take into account that the kernel matrix can be computed in a fully parallel way, thus, at least in theory, all the learning process can be completely parallelized.

So the advantage of this approach mainly consists in the intrinsic parallel processing. This technique can be very useful in ‘early learning’ problems, requiring that a machine exhibits an extremely fast learning performance. Whenever the kernel matrix changes, modifications can be mapped via reprogramming of resistive network. The following Section describes the procedure to map Support Vector Machines to a proper co-content network.

4.2.3 Hardware SVM Training

4.2.3.1 Circuit design strategy

The circuit complexity brought about by the linear constraint in (2.97) is handled by a reformulation of the SVM cost function. Theory shows [63] that, in the presence of a positive definite kernel function, the linear constraint in eq. (2.97) can be removed without affecting the generalization ability of the SVM. Indeed, forcing a null bias term b in the decision function implies to take out the linear constraint in the dual problem formulation (2.103). This rewriting modifies the SVM functional as per (2.103)

Such a simplified formulation of the original problem holds for positive

definite kernels. From a circuit perspective, it is interesting to anticipate that the “box constraints” on the parameters α prevent the parameters themselves to reach unfeasible values; in other words, both constraints imply useful physical limitations on the voltage values of the associate vector coordinates \mathbf{v} . The resulting design of a circuit mapping a functional, $\mathfrak{S}(\mathbf{a})$, into its co-content potential, $G(\mathbf{v})$, follows two basic steps:

1. Setting up a correspondence between vectors \mathbf{a} and \mathbf{v} (association step);
2. Setting up a topological structure for \mathfrak{R} (circuit definition)

4.2.3.2 Association step

First, one sets a reference voltage, V_0 , to map SVM parameters α_i into voltages v_i , hence the relation between SVM variables and voltages is defined as $\alpha_i = v_i/V_0$. Thus, the box constraint takes the form: $0 \leq v_i \leq \Omega V_0 \quad i = 1, \dots, n$, while the quadratic functional can be rewritten as

$$\frac{1}{2} \boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha} - \sum_{i=1}^n \alpha_i \quad \rightarrow \quad \frac{1}{2V_0^2} \mathbf{v}^t \mathbf{Q} \mathbf{v} - \frac{1}{V_0} \sum_{i=1}^n v_i \quad (4.15)$$

By choosing an arbitrary resistance value, R_0 , and multiplying the dimensionless functional (4.15) by the power reference V_0^2/R_0 , one defines the actual co-content potential function $G(\mathbf{v})$, as

$$\Gamma(v) = \frac{1}{2R_0} \mathbf{v}^t \mathbf{Q} \mathbf{v} - \frac{V_0}{R_0} \sum_{i=1}^n v_i \stackrel{def}{=} \Psi(\mathbf{v}) - \Phi(\mathbf{v}) \quad (4.16)$$

The physical dimension of both Ψ and Φ is power, and is consistent with the co-content potential function terms. By taking the partial derivatives of Ψ with respect to voltages v_i , one obtains the current terms:

$$\hat{i}_k = \frac{\partial \Psi}{\partial v_k} = \frac{1}{R_0} \sum_{i=1}^n q_{ki} v_i \quad (4.17)$$

Likewise, partial derivatives of Φ yield:

$$\tilde{i}_k = \frac{\partial \Phi}{\partial v_k} = -\frac{V_0}{R_0} \quad (4.18)$$

Finally by assembling $\nabla\Gamma(\mathbf{v})$ one obtains total currents; in particular, the current at the k -th terminal is:

$$i_k = \hat{i}_k + \tilde{i}_k. \quad (4.19)$$

4.2.3.3 Circuit definition

A constant current source can easily support the second term \tilde{i}_k in the expression (4.19). The first term \hat{i}_k , because \mathbf{Q} matrix is positive definite, can be implemented by a reciprocal circuit $\hat{\mathfrak{R}}$ containing two-terminal resistors only. The topological structure is not unique. A possible choice is sketched in 4.7 for the very simple case $n = 3$. Following [159], the general form of the $n \times n$ conductance matrix G is:

$$G = \begin{pmatrix} \sum_k y_{1k} & -y_{12} & \cdots & \cdots & -y_{1(n-1)} & -y_{1n} \\ -y_{21} & \sum_k y_{2k} & \cdots & \cdots & -y_{2(n-1)} & -y_{2n} \\ -y_{31} & \cdots & \cdots & \cdots & \cdots & -y_{3n} \\ -y_{41} & -y_{42} & \cdots & \cdots & \cdots & -y_{4n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ -y_{n1} & -y_{n2} & \cdots & \cdots & \cdots & \sum_k y_{n,k} \end{pmatrix} \quad (4.20)$$

with $y_{ik} = y_{ki}$ according to the well-known reciprocity theorem. The correspondence between the y_{ij} elements and the elements q_{ij} of \mathbf{Q} is

$$y_{ij} = \begin{cases} -\frac{q_{ij}}{R_0}; & i \neq j \\ \frac{\sum_k q_{ik}}{R_0}; & i = j \end{cases} \quad (4.21)$$

as one can find after some manipulations. As a final step, the circuit implementation of the n box constraints is obtained through nonlinear resistors with the $v_d - i_d$ characteristic in 4.8.

As shown in [159], these resistors do not contribute to the co-content of $\hat{\mathfrak{R}}$. The structure of $\hat{\mathfrak{R}}$, resulting by connecting the multiterminal $\hat{\mathfrak{R}}$ defined by (4.21) to these nonlinear resistors and to the current generators (4.18) is shown in 4.9, where a set of identical linear capacitors completes the circuit.

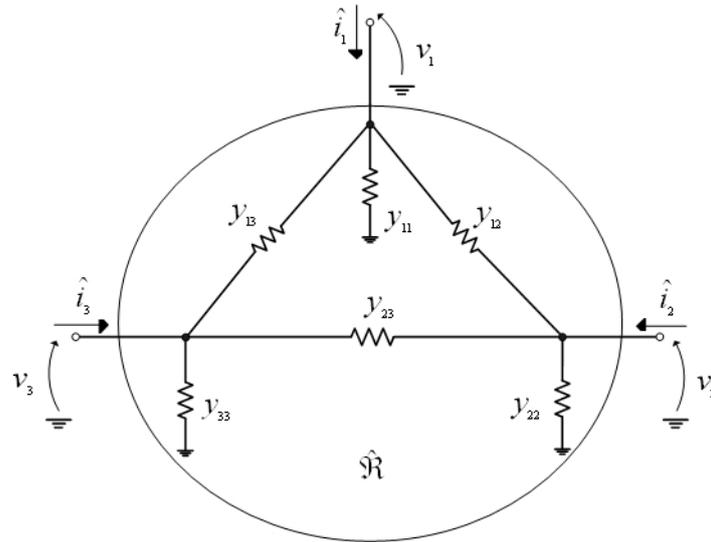


Figure 4.7: Circuit topology: three terminal case

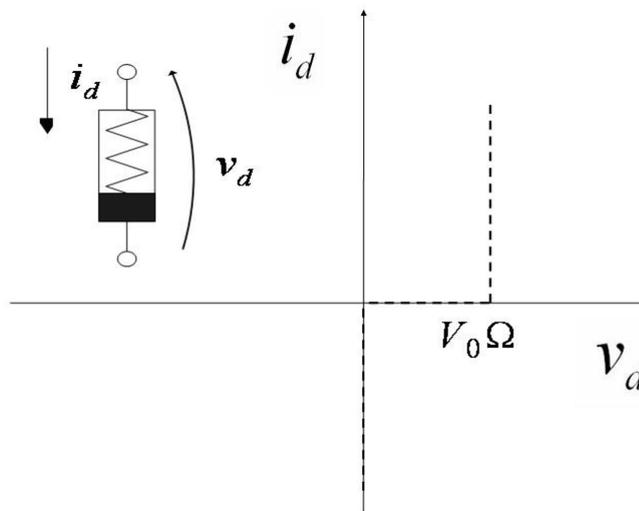


Figure 4.8: Voltage limiting component and voltages-current curve

4.2.3.4 Complexity reduction

A crucial aspect of the overall approach concerns the structure of the resistive network, since the complexity of \mathfrak{R} depends on the density of the kernel matrix. Using the popular linear or RBF kernel, for instance, tends to yield a dense Hessian matrix \mathbf{Q} , and ultimately complicates the topology of the supporting network, \mathfrak{R} . Generally speaking, the number of two-terminal resis-

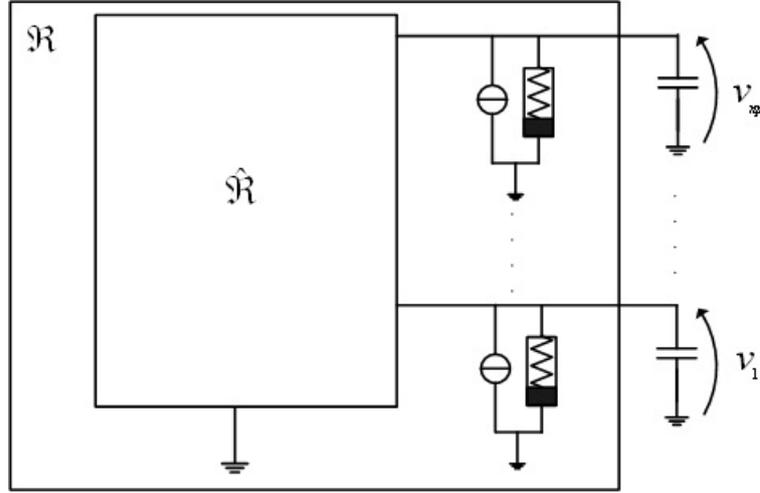


Figure 4.9: Complete circuit

tors implementing a full-density matrix Q is equal to $n(n+1)/2$. By contrast, a proper choice of the kernel function can lead to a sparse Q matrix, and reduce the complexity of R accordingly.

In general, taking a positive-definite kernel matrix and annihilating its closest-to-zero elements yields a kernel matrix which is no longer positive definite. In other words, the sparse structure of a positive definite matrix must be guaranteed ab origine by a proper choice of the kernel.

A positive definite kernel generating a sparse Q matrix has been defined in [162]. More specifically, thanks to this kernel the sparsity of Q can be tuned through a specific parameter (cut-off distance), according to the following definition.

Let $d(\mathbf{x}_l, \mathbf{x}_m) = \|\mathbf{x}_l - \mathbf{x}_m\|^2$ denote the distance between the pair of points $(\mathbf{x}_l, \mathbf{x}_m)$, and let r be a positive cut-off distance. The kernel-based inner product, $K(\mathbf{x}_l, \mathbf{x}_m)$, is set to 0 each time $d(\mathbf{x}_l, \mathbf{x}_m) \geq 2r$; otherwise, the quantity $K(\mathbf{x}_l, \mathbf{x}_m)$ is worked out by means of a recursive procedure:

$$\begin{aligned}
 k_{d,1}(\mathbf{x}_l, \mathbf{x}_m) &= 1 - \frac{\|\mathbf{x}_l - \mathbf{x}_m\|}{2r} \\
 k_{d,2}(\mathbf{x}_l, \mathbf{x}_m) &= \arccos\left(\frac{\|\mathbf{x}_l - \mathbf{x}_m\|}{2r}\right) - \frac{\|\mathbf{x}_l - \mathbf{x}_m\|}{2r} \sqrt{1 - \left(\frac{\|\mathbf{x}_l - \mathbf{x}_m\|}{2r}\right)^2} \\
 k_{d,j}(\mathbf{x}_l, \mathbf{x}_m) &= \frac{j-1}{j} k_{d,j-2}(\mathbf{x}_l, \mathbf{x}_m) - \frac{1}{j} \frac{\|\mathbf{x}_l - \mathbf{x}_m\|}{2r} \left(1 - \left(\frac{\|\mathbf{x}_l - \mathbf{x}_m\|}{2r}\right)^2\right)^{\frac{j-1}{2}}
 \end{aligned} \tag{4.22}$$

Recursions stop when the running index j reaches the dimension n of the

original data space, so one has: $K(\mathbf{x}_l, \mathbf{x}_m) = k_{d,d}(\mathbf{x}_l, \mathbf{x}_m)$. As shown in [162], this kernel is positive definite and ensures good prediction performances while yielding sparse kernel matrices.

4.2.4 Experimental Results

The circuit capabilities have been tested through a set of experimental sessions on three non toy known test sets from UCI repository [100]: Sonar, Ionosphere and Diabetes. The assessment of the generalization ability compared the method's performances with those attained by a software-based, high-precision SVM implementation, including a bias term. To measure the effect of using a sparse kernel matrix, different levels of the sparsity-controlling parameter were tested.

The software implementation adopted a conventional SMO-based algorithm, including Lin's first-order heuristic to select a working set [62]. In all experiments, the tolerance on the KKT conditions was 10^{-3} . Circuit-based optimization runs were accomplished first by generating the components netlist in an automated fashion, then by applying Spice (Orcad 16.0) simulations.

The hardware simulations of the voltage-limiting section involved a classic diodes-based limiting circuitry. The model used was the default diode with a constant threshold of $0.594V$; this voltage value tuned the offset to get a precise box constraint when the parameter-representative voltage was 0 or the SVM bounding constant, Ω . Additional circuit-design parameters were set as follows: $C = 1nF$, $V_0 = 1V$, $R_0 = 1k\Omega$. For a given $\hat{\mathfrak{R}}$, the values of C and R_0 control the rate of convergence of the dynamic process in the circuit. All solution voltages values were measured upon completion of a transient interval of $80\mu s$. In all the cases, the circuit converged to the solution without instabilities, as expected.

The domain of kernel parameter r was set as $[0.2, 4]$ and all attributes of data were normalized in $[-1, +1]$. Table 4.9 gives, for the various data sets, the splitting strategies of the data into training and test sets, and the associate values of the SVM regularization parameter Ω .

Tables 4.10, 4.11, 4.12 report on the test-set accuracy values and the com-

Dataset	#Training	#Test	Ω
Sonar	150	58	1
Ionosphere	251	100	8
Diabetes	230	538	1

Table 4.9: Datasets Splitting and Ω parameters

plexity values for the hardware SVM together with the values obtained with SVM software with bias; 4.10, 4.11, 4.12 are the graphical counterparts of the tables.

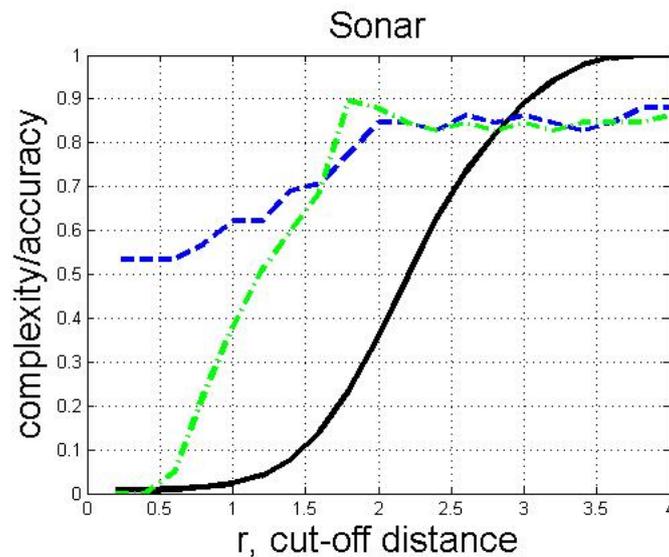


Figure 4.10: Sonar dataset: dashed line represents the software accuracy, continuous line is the circuitual complexity and dashed-dotted line is svm hardware accuracy

From an accuracy viewpoint, results suggest that the training circuitry reaches a satisfactory solution in all three cases.

However the most interesting aspect here is that a good level of accuracy ($>75\%$) can be maintained also when the complexity of the resistor network \mathfrak{R} is severely limited through a proper value of the cut-off distance parameter r .

In particular:

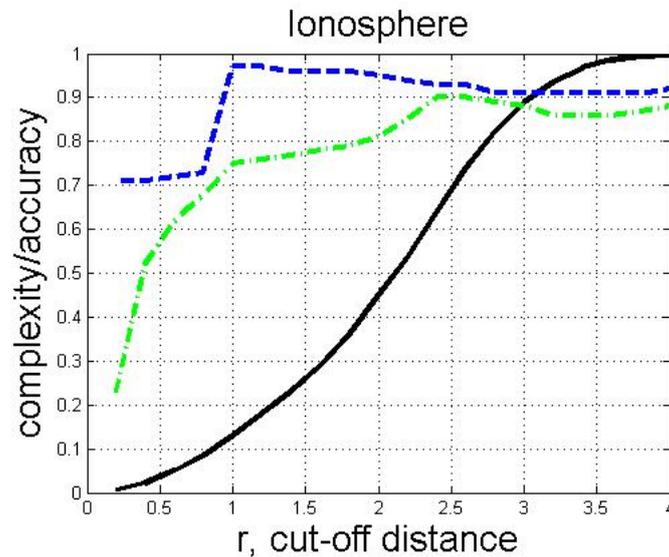


Figure 4.11: Ionosphere dataset: dashed line represents the software accuracy, continuous line is the circuitual complexity and dashed-dotted line is svm hardware accuracy

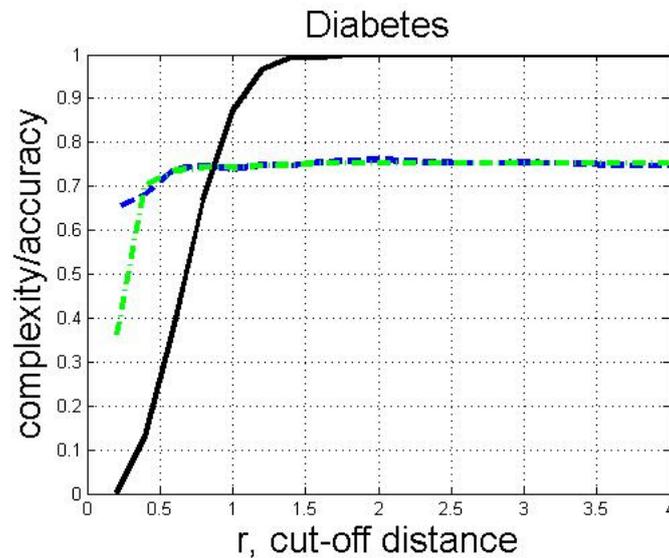


Figure 4.12: Diabetes dataset: dashed line represents the software accuracy, continuous line is the circuitual complexity and dashed-dotted line is svm hardware accuracy

1. in the case of Sonar dataset with a sparsity level of 80% the accuracy is unchanged respect the full matrix training

2. in Ionosphere an acceptable accuracy is obtained at a level of sparsity of 80%
3. in Diabetes a level of sparsity of 70% can guarantee the maximum accuracy obtainable with full matrix

Note that the degrading results obtained when sparsification is near 100% are due to the absence of the bias term b . For $b=0$ the decision function $\text{sign}(w^T x_i + b)$ can take uncorrect values when $w^T x_i$ is close to zero. In these cases, for a fair evaluation, the result has been considered as an error. Then the accuracy results reported here must be interpreted as the most conservative evaluations.

In general the results discussed here appear to be quite encouraging. However it seems to remain a certain dependence of the effectiveness of the sparsification from the chosen dataset. It should be also underlined that the proposed approach can use other sparse kernel representations and that, in every case, the network is able in giving a coherent solution in both the case of sparse or not sparse network.

4.2.5 Conclusions

In this work a modification of SVM formulation to make it compatible with a co-content minimization network implementation is proposed; in particular bias removal allowed eliminating the linear constraint term.

Further the circuit complexity was strongly reduced and controlled by using an effective sparse kernel function. Experimental evidence on non toy problems confirmed that the circuit approach attained accuracy classification levels comparable with those yielded by high precision software implementation.

The learning skill of the network depends on the value of the kernel parameter, hence a future line of investigation concerns the definition of an operative criterion of selection of r . Another possible extension regards the application of other possible sparse kernels presented in literature and the comparison of the obtained results with those obtained in the present case. As

r	Complexity	SW SVM	HW SVM
0.2	0.007	0.534	0
0.4	0.007	0.534	0
0.6	0.0098	0.534	0.05
0.8	0.014	0.569	0.224
1	0.022	0.621	0.379
1.2	0.04	0.621	0.51
1.4	0.077	0.69	0.6
1.6	0.141	0.707	0.689
1.8	0.236	0.776	0.896
2	0.358	0.845	0.879
2.2	0.496	0.845	0.845
2.4	0.628	0.828	0.828
2.6	0.735	0.862	0.845
2.8	0.819	0.845	0.828
3	0.891	0.862	0.845
3.2	0.942	0.845	0.828
3.4	0.977	0.828	0.845
3.6	0.9944	0.845	0.845
3.8	0.9988	0.879	0.845
4	1	0.879	0.862

Table 4.10: Comparison of SW SVM and HW SVM accuracy for Sonar Dataset

r	Complexity	placeSW SVM	HW SVM
0.2	0.007	0.71	0.23
0.4	0.022	0.71	0.52
0.6	0.05	0.72	0.62
0.8	0.0851	0.73	0.68
1	0.13	0.97	0.75
1.2	0.18	0.97	0.76
1.4	0.23	0.96	0.77
1.6	0.29	0.96	0.78
1.8	0.36	0.96	0.79
2	0.45	0.95	0.81
2.2	0.5351	0.94	0.85
2.4	0.638	0.93	0.9
2.6	0.74	0.93	0.9
2.8	0.821	0.91	0.89
3	0.89	0.91	0.88
3.2	0.936	0.91	0.86
3.4	0.967	0.91	0.86
3.6	0.986	0.91	0.86
3.8	0.9945	0.91	0.87
4	0.998	0.92	0.88

Table 4.11: Comparison of SW SVM and HW SVM accuracy for Ionosphere Dataset

r	Complexity	placeSW SVM	HW SVM
0.2	0.002	0.651	0.363
0.4	0.131	0.682	0.703
0.6	0.385	0.74	0.736
0.8	0.67	0.747	0.742
1	0.87	0.74	0.742
1.2	0.966	0.749	0.747
1.4	0.9931	0.747	0.749
1.6	0.995	0.755	0.753
1.8	1	0.757	0.753
2	1	0.76	0.753
2.2	1	0.757	0.753
2.4	1	0.755	0.753
2.6	1	0.753	0.753
2.8	1	0.753	0.753
3	1	0.755	0.753
3.2	1	0.753	0.753
3.4	1	0.751	0.753
3.6	1	0.749	0.753
3.8	1	0.747	0.753
4	1	0.747	0.753

Table 4.12: Comparison of SW SVM and HW SVM accuracy for Diabetes Dataset

a final step a concrete prototypal VLSI implementation of the network could be developed.

4.3 Fast Approximate Regularized Least Squares

Large-scale learning represents a crucial topic in the research area of Machine Learning, and kernel methods are of particular interest for non-linear learning. Different approaches have been proposed to address such issue. Sequential Minimization Optimization (SMO) for Support Vector Machines (SVMs) [62], and the conjugate gradient method [55] represent well-known techniques that can prove useful for large-scale problems when high dimensional spaces are involved. This work introduces a non-iterative method for the approximate solution of large-scale learning. The Regularized Least Squares (RLS) [53] framework supports the learning principle. The rationale behind this choice is that RLS is a well-known and successful Machine-Learning algorithm, whose training procedure consists in solving a system of linear equations. Efficient solvers exist for the RLS paradigm; however, they cannot address effectively large-scale problems. Solvers based on decomposition methods [62] do not allow one to predict execution time or complexity easily. Direct-solution methods, which combine Gaussian elimination with a matrix factorization technique (Cholesky decomposition) [163], exhibit predictable time and complexity, but suffer from two major drawbacks: first, the whole system matrix must be kept, hence storage requirement scales as $O(n^2)$, where n is the number of rows/columns of the linear system; secondly, the solver complexity scales with $O(n^3)$. Iterative methods [55] can outperform in speed the latter approaches in the presence of sparse matrixes, but still require the computation of the whole matrix; moreover, the speed-up is not predictable and heavily depends on the specific problem settings.

When tackling large-scale problems or when using devices with limited resources, one requires approximation schemes that can grant a trade-off between accuracy and computational complexity. Toeplitz matrix [164] are particularly interesting, as the solution of a Toeplitz system with n variables reduces computational complexity to $O(n^2)$ and storage requirements to $O(n)$. Toeplitz solvers have been successfully used in Linear Predictive Coding, and in general where a Toeplitz system emerges, as in autocorrelation-based methods [165].

This work exploits the properties of Toeplitz matrixes to significantly reduce both the computational cost and the memory space required to train an RLS-based machine.

The research presented in this section first derives a sufficient condition that remaps the RLS learning problem into a Toeplitz system for one-dimensional regression problems; then, an approximation scheme for multivariate data is proposed. This general scheme balances efficiency versus accuracy and can be used to address large-scale classification problems effectively; indeed, it is showed that approximation accuracy is high as long as the kernel function leads to a kernel matrix that is very close to having a Toeplitz-based structure. The remarkable reduction of storage requirements and the exact predictability of memory usage represents in particular the crucial advantage provided by the present framework when compared with other approaches proposed in the literature. one should also consider that the method scales in memory as $O(n)$ using at the same time all the available patterns. That feature makes actually the framework also amenable for limited-resource implementations involving embedded systems [166].

4.3.1 Toeplitz Matrixes for Regularized Least Squares

A Toeplitz matrix, \mathbf{T} , of size $n \times n$ is a diagonal-constant matrix. If the matrix is symmetric, n values completely specify \mathbf{T} ; thus, if one denotes with $\mathbf{k} \in \mathbb{R}^n$ the vector that contains the elements duplicated in each diagonal, one verifies that: $T_{i,j} = k_{|i-j|}$. With these specifications, \mathbf{k} spans the first row of the Toeplitz matrix, and the matrix takes the form:

$$\begin{pmatrix} k_0 & k_1 & k_2 & \dots & k_{n-2} & k_{n-1} \\ k_1 & k_0 & k_1 & \dots & \dots & k_{n-2} \\ \vdots & k_1 & k_0 & \dots & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ k_{n-2} & \dots & \dots & \dots & \dots & k_1 \\ k_{n-1} & k_{n-2} & \dots & \dots & k_1 & k_0 \end{pmatrix} \quad (4.23)$$

Theory shows that any system of equations supported by a Toeplitz ma-

trix, \mathbf{T} , expressed as $\mathbf{T}\beta = \mathbf{y}$ can be solved efficiently by the Levinson-Trench-Zohar (LTZ) recursive algorithm [164]. The advantage of such an approach is that the complexity of the solution process scales as $O(n^2)$ in time and as $O(n)$ in memory, hence it outperforms any Gaussian elimination method. These features make the LTZ algorithm very appealing when one needs an efficient approach to solve a Toeplitz system.

The LTZ algorithm in Matlab code is the following:

Algorithm 8 Symmetric Toeplitz Solver

Require: \mathbf{r} , first row/column of matrix \mathbf{T} , labels vector \mathbf{y}

Ensure: solution vector \mathbf{x}

```

1: function  $\mathbf{x} = \text{ToeplitzSolver}(\mathbf{r}, \mathbf{y})$ 
2:  $N = \text{length}(\mathbf{r})$ ;
3:  $\mathbf{a} = [1]$ ;
4:  $\mathbf{b} = [1]$ ;
5:  $\text{eps} = \mathbf{r}(1)$ ;
6:  $\mathbf{x} = [\mathbf{y}(1)/\text{eps}]$ ;
7:  $\mathbf{r} = \mathbf{r}(:)'$ ;
8:  $\mathbf{y} = \mathbf{y}(:)$ ;
9: for  $n=2:N$  do
10:    $\text{subr} = \mathbf{r}(n:-1:2)$ ;
11:    $\text{vareps} = -(1/\text{eps}) * \text{sum}(\text{subr} * \mathbf{a})$ ;
12:    $\text{csubr} = \text{subr}(\text{end}:-1:1)$ ;
13:    $\text{nu} = -(1/\text{eps}) * \text{sum}(\text{csubr} * \mathbf{b})$ ;
14:    $\text{tempa} = \mathbf{a}$ ;
15:    $\text{tempb} = \mathbf{b}$ ;
16:    $\mathbf{a} = [\text{tempa} \ 0] + \text{vareps} * [0 \ \text{tempb}]$ ;
17:    $\mathbf{b} = [0 \ \text{tempb}] + \text{nu} * [\text{tempa} \ 0]$ ;
18:    $\text{eps} = \text{eps} * (1 - \text{vareps} * \text{nu})$ ;
19:    $\text{lambda} = \mathbf{y}(n) - \text{sum}(\text{subr} * \mathbf{x})$ ;
20:    $\mathbf{x} = [\mathbf{x} \ 0] + (\text{lambda}/\text{eps}) * \mathbf{b}$ ;
21: end for
22:  $\mathbf{x} = \mathbf{x}'$ ;

```

In the following, a theoretical analysis derives a sufficient condition to formulate RLS training in terms of a Toeplitz matrix for one-dimensional problems, and an approximation scheme that, starting from a general kernel matrix \mathbf{K} , yields a Toeplitz kernel \mathbf{T} for multidimensional domains.

4.3.1.1 Toeplitz Kernels for Univariate RLS problems

Toeplitz matrixes naturally emerge when dealing with translation-invariant kernels and uniform (step-constant) sampling of univariate data. The following Lemma relates one-dimensional data distributions to Toeplitz matrixes.

Lemma 4.3.1. *Given a set, \mathbf{X} of mono-dimensional patterns drawn from uniform sampling and a translation-invariant kernel such that: $K(u, v) = K(\|u - v\|)$, then the associate kernel matrix is a Toeplitz (symmetric) matrix.*

Proof: The assertion is proved by construction. Because of the uniform sampling, one can write the j -th sample as $x_j = x_0 + \Delta j$, where x_0 is the first sample of the set and Δ is the sampling step. To compute the kernel function one works out the element:

$$K_{i,j} = K(x_i, x_j) = K(|x_0 + i\Delta - (x_0 + j\Delta)|) = K(|i - j| \Delta) \tag{4.24}$$

In matrix form, this becomes:

$$\begin{pmatrix} K(0) & K(\Delta) & \dots & \dots & K((n-2)\Delta) & K((n-1)\Delta) \\ K(\Delta) & K(0) & K(\Delta) & \dots & \dots & K((n-2)\Delta) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ K((n-1)\Delta) & \dots & \dots & \dots & \dots & K(0) \end{pmatrix} \tag{4.25}$$

which is a Toeplitz matrix; this proves the assertion. **Q.E.D.**

It is easy to verify that, for any kernel matrix of Toeplitz type, \mathbf{T} , the system $(\mathbf{T} + \lambda I_{nn})\beta = \mathbf{y}$ is of Toeplitz type as well, hence a fast solution with the LTZ

algorithm is attainable. This proves that regression problems can be solved exactly and rapidly by the RLS approach when dealing with univariate, uniformly sampled large training sets. The work by [45] provides an example of application of Lemma 4.3.1.

4.3.1.2 Approximated Toeplitz Systems for Multivariate RLS Problems

Tackling multivariate data makes it difficult, or even impossible, to set a sufficient condition ensuring that the kernel matrix always is in Toeplitz form. To benefit from the LTZ algorithm, one might yet approximate the original $n \times n$ kernel matrix, \mathbf{K} , by its nearest Toeplitz approximation, $\mathbf{T}_{\mathbf{K}}$; the similarity between matrixes is ruled by a specified metric measure, M . This approach requires one to solve the following problem:

$$\min_{\mathbf{T}_{\mathbf{K}}} \|\mathbf{T}_{\mathbf{K}} - \mathbf{K}\|_M \quad (4.26)$$

with the constraint that $\mathbf{T}_{\mathbf{K}}$ is positive semi-definite. To solve (4.26) one usually applies iterative algorithms that prove computationally expensive [167].

The research presented in the following yields an efficient approach that also gives an effective approximation scheme. The problem (4.26) has an analytical solution if one relaxes the constraint on the positive semi-definite property of $\mathbf{T}_{\mathbf{K}}$, and the proximity is measured by the Frobenius norm, $\|\cdot\|_F$, of the difference matrix:

$$\|\mathbf{T}_{\mathbf{K}} - \mathbf{K}\|_F = \|\mathbf{D}\|_F = \sqrt{\sum_i \sum_j |d_{ij}|^2} \quad (4.27)$$

Lemma 4.3.2. *Given a set X of multivariate samples the solution of $\min_{\mathbf{T}_{\mathbf{K}}} \|\mathbf{T}_{\mathbf{K}} - \mathbf{K}\|_F$ is attained by the matrix $\mathbf{T}_{\mathbf{K}}$ that is built by setting all elements of each constant-value diagonal to the mean value of the corresponding diagonal in \mathbf{K} .*

Proof: First, one unrolls in a diagonal-wise fashion the matrix \mathbf{K} , such that each diagonal is concatenated to each other. Considering the symmetry of \mathbf{K} one only concatenates the upper diagonals; indicating by d_i each diagonal the resulting vector $\mathbf{v}_{\mathbf{K}}$ is of length $(n(n+1))/2$.

$$\mathbf{v}_K = \left\{ \underbrace{K_{0,0}, K_{1,1}, \dots, K_{n-1,n-1}, \dots}_{d_0}, \underbrace{K_{0,j}, K_{1,j+1}, \dots, K_{n-1-j,n-1}, \dots}_{d_j}, \underbrace{K_{0,n-1}}_{d(n-1)} \right\}$$

Then one performs the same unrolling procedure for the Toeplitz matrix, \mathbf{T}_K , and builds \mathbf{v}_T

$$\mathbf{v}_T = \left\{ \underbrace{T_{0,0}, T_{0,0}, \dots, T_{0,0}, \dots}_{d_0}, \underbrace{T_{0,j}, T_{0,j}, \dots, T_{0,j}, \dots}_{d_j}, \underbrace{T_{0,n-1}}_{d(n-1)} \right\}$$

the difference vector $\delta = \mathbf{v}_K - \mathbf{v}_T$ is:

$$\delta = \left\{ \underbrace{K_{0,0} - T_{0,0}, K_{1,1} - T_{0,0}, \dots, K_{n-1,n-1} - T_{0,0}, \dots}_{d_0}, \underbrace{K_{0,j} - T_{0,j}, K_{1,j+1} - T_{0,j}, \dots, K_{n-1-j,n-1} - T_{0,j}, \dots}_{d_j}, \underbrace{K_{0,n-1} - T_{0,n-1}}_{d(n-1)} \right\}$$

Finally the problem $\min_{\mathbf{T}_K} \|\mathbf{T}_K - \mathbf{K}\|_M$ reduces to find $\min_{\mathbf{T}_K} \|\delta\|_2^2$ where

$$\|\delta\|_2^2 = \left\{ \sum_{j=0}^{n-1} \underbrace{\sum_{i=0}^{n-1-j} (K_{i,j+i} - T_{0,j})^2}_{d_j} \right\}. \text{ All terms in the summation are positive or zero, therefore the minimum is attained when each term in round brackets is minimum: however due to the Toeplitz diagonal-constant structure one has to consider together all the terms within each diagonal. Given a diagonal } j \text{ then the minimum of } \underbrace{\sum_{i=0}^{n-1-j} (K_{i,j+i} - T_{0,j})^2}_{d_j}, \text{ is attained when } T_{0,j} \text{ is}$$

the sample mean of the diagonal j of the original matrix \mathbf{K} . This observation holds for each diagonal. **Q.E.D.**

offers two principal advantages: first, the solution of problem (4.26) is expressed analytically; secondly, to work out \mathbf{T}_K one should only compute the

mean values of each diagonal of the matrix \mathbf{K} . Once the mean value of a diagonal is computed, the associated memory can be de-allocated and re-used; this leads to a memory occupation that scales as $O(n)$.

The approximation scheme based on is most effective when the original kernel matrix \mathbf{K} , has an "almost-Toeplitz" structure, so that alterations in the diagonal elements marginally distort the overall information carried by K . To evaluate the accuracy attained by $\mathbf{T}_{\mathbf{K}}$ and to measure the distortion brought about by the approximation, one should ultimately estimate the error performed on a test set.

The relaxation of the constraint on the positive semi-definiteness of $\mathbf{T}_{\mathbf{K}}$ is of minor importance in practice, for several reasons. First, even if $\mathbf{T}_{\mathbf{K}}$ is indefinite (that is a matrix neither positive nor negative semi-definite), the regularization parameter, $\lambda > 0$, that is added to each term in the main diagonal contributes to make the matrix positive semi-definite (singularities are in fact very unlikely). Secondly, if the matrix $[\mathbf{T}_{\mathbf{K}} + \lambda \mathbf{I}_{nn}]$ still results indefinite, it can be shown that the associated RLS training problem is equivalent to learning in a Reproducing Kernel Krein Space [52]. Theory shows [52] that learning is possible in such a space, and Rademacher bounds to the generalization error can be estimated accordingly. Finally, the Look-Ahead-Levinson algorithm [168], which is a version of the LTZ method, can effectively deal with indefinite Toeplitz matrixes almost without extra costs.

The advantage of the Toeplitz-based approach becomes apparent when one considers the analysis summarized in 4.13, where the features of the conventional and of the Toeplitz-based solutions of the RLS learning problem are compared (for the conjugate gradient a underestimation of the computational cost is given). The crucial aspect is that the sharp reduction in all requirements makes the direct-solution approach viable for large data sets, which would otherwise prove inaccessible.

4.3.1.3 Effects of RBF Kernels on Generalization

The distortion introduced by the Toeplitz kernel, $\mathbf{T}_{\mathbf{K}}$, in approximating \mathbf{K} affects the accuracy of the classifier machine. Therefore, the kernel function should ensure a satisfactory trade-off between accuracy and computational

Requirements	Gaussian Eli.	Conj. Gradient	Toeplitz
Computational Complexity	$O(n^3)$	$O(kn^2)$	$O(n^2)$
Storage Requirement	$O(n^2)$	$O(n^2)$	$O(n)$

Table 4.13: Comparison among Gaussian Elimination, Conjugate Gradient and the Toeplitz-based solver approaches to RLS learning: n is the number of patterns and k is the number of CG iterations

complexity. This section shows that the Radial-Basis-Function (RBF) kernel can accomplish this requirement and is suitable for Toeplitz approximations.

The kernel formulation, $K(\mathbf{u}, \mathbf{v}) = \exp(-\|\mathbf{u} - \mathbf{v}\|^2 / (2\sigma^2))$, implies that inner products vary in the range $(0, 1]$. By varying the kernel parameter, σ , one can drive the numerical distribution of similarity results towards either extremum of the range. Let $\phi(\mathbf{u})$, $\phi(\mathbf{v})$ the non-linear mappings induced by the Gaussian kernel on pattern vectors \mathbf{u} , \mathbf{v} , respectively; using kernels properties one has $\|\phi(\mathbf{u}) - \phi(\mathbf{v})\|^2 = K(\mathbf{u}, \mathbf{u}) - 2K(\mathbf{u}, \mathbf{v}) + K(\mathbf{v}, \mathbf{v})$ then because one is using RBF kernel one has:

$K(\mathbf{u}, \mathbf{u}) - 2K(\mathbf{u}, \mathbf{v}) + K(\mathbf{v}, \mathbf{v}) = 2[1 - K(\mathbf{u}, \mathbf{v})]$. Two extreme situations can be represented as follows 4.13:

1. $\sigma \rightarrow 0$: then $K(\mathbf{u}, \mathbf{v}) \rightarrow 0 \forall \mathbf{u}, \mathbf{v}$; thus all distances in the Hilbert space collapse to a constant $\|\phi(\mathbf{u}) - \phi(\mathbf{v})\|^2 = 2 \cdot [1 - K(\mathbf{u}, \mathbf{v})] \cong 2$. The images of all patterns in the infinite-dimensional space lay on a hyper-sphere, and the kernel matrix tends to the identity matrix.
2. $\sigma \rightarrow \infty$, then $K(\mathbf{u}, \mathbf{v}) \rightarrow 1 \forall \mathbf{u}, \mathbf{v}$; all distances in the feature space collapse to $\|\phi(\mathbf{u}) - \phi(\mathbf{v})\|^2 = 2 \cdot [1 - K(\mathbf{u}, \mathbf{v})] \cong 0$. All images collapse onto one point and the entries in the kernel matrix are all set to 1.

In both those situations, one expects the resulting generalization performance to be far from optimal, due to over-fitting in the former case, and over-smoothing in the latter. It is however intriguing that, in both cases, the kernel matrix yet tends to a Toeplitz matrix. The case $\sigma \rightarrow 0$ may be especially interesting because images do not concentrate in one point of the kernel space,

and the key issue is to set σ to a value that is small enough to drive \mathbf{K} toward a Toeplitz matrix and, at the time, to ensure that over-fitting is not severe. From another perspective also the case $\sigma \rightarrow \infty$ can be interesting because it is well known, that for ‘big enough’ values of σ one recover the linear kernel [169]. Thus the Gaussian kernel provides a suitable kernel function which gives an effective generalization ability and also benefits from a Toeplitz-like structure matrix.

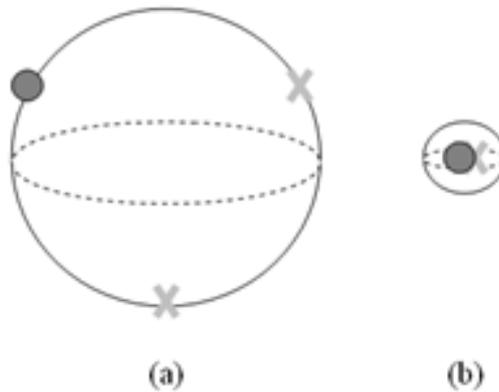


Figure 4.13: Kernel space data representation (crosses are +1 data and balls are -1 data): (a) is the case $\sigma \rightarrow 0$, (b) is the case $\sigma \rightarrow \infty$

4.3.2 Experimental Results

The experimental session of the presented research aimed at:

1. checking the consistency of the approximated solution
2. assessing the gap in accuracy between classical solvers and the approximated one
3. evaluating the average speed-up attained by the Toeplitz-based approach.

To achieve those goals, the proposed framework was tested on several classification problems, and compared with both a conventional solver [163] and a solver based on the conjugate-gradient method [55]. As the approximation scheme may coarsely alter some elements of the original kernel matrix,

Dataset	#Training Set	#Test Set	# Variables
MNist3vs8	1,5,10,20	4	80
Ijcnn	1,5,10,20,50,100	5	22
Daimler	1,2,5,7	1	648
w8a	1,5,10,20,40	5	300
Covertypes	1,5,10,20,50,100	5	54
Faces	1,5,10,20	1	361

Table 4.14: Data splitting criteria for the data sets used in the experiments and number of variables. The numbers of patterns in the table are intended multiplied by $1e3$

classification problems represent a suitable applicative domain for a learning strategy that addresses a trade-off between accuracy and complexity.

Experimental verifications involved real-world, binary datasets, namely: Covertypes, Ijcnn, w8a [62], Daimler [170], Manuscript NIST (3 vs 8) [99], and MIT face database [171]. In all cases, the patterns were shuffled, each coordinate was normalized into the range $[-1, +1]$, and each data set was randomly split into a training set and a test set. Table 4.14 gives the partitioning criteria and the number of variables for each dataset in the experiments. For each dataset, a grid-based model selection method was used to tune the parameters; in the following, σ^C and σ^T will denote the best kernel parameter for the classical and the Toeplitz-based approach, respectively. Optimal settings for parameters λ and σ have been set by adopting a conventional cross-validation strategy based on a test set.

The tests have been implemented in a Matlab 2009a environment and on a Intel Xeon 5440 Quadcore. The standard Matlab Gaussian Elimination solver (`/` operator) and the standard conjugate gradient solver without preconditioning (function `pcg`) of the Matlab optimization tool box have been used. The first is a built-in function, the second is a .m file that can be considered almost a built-in function due to the fact that all the expensive operations are matrix multiplications implemented by the fast built-in `mtimes` function.

Both CG and Gaussian Solver are parallel. CG was used by exploiting the default stopping criterion and by setting 100 as the maximum number iterations. The LTZ algorithm for the approximated solution was a mex file C-coded with no parallelism.

4.3.2.1 Result Analysis

The graph in 4.14 is a preliminary experiment and shows the behavior of the normalized Frobenius norm $\|\mathbf{T}_K - \mathbf{K}\|_F / n^2$ with changing kernel parameter $\log(\sigma)$. As reference the MNIST dataset and $n = 1e3$ samples were used; as theory showed the best approximating behavior verifies for both low and high values of σ ; analogous results hold for other datasets.

The graphs in 4.15 give, for each testbed, the classification error scored by the 'best' model on the test set, for a varying number of training patterns. In each graph, the solid black line gives the best test error attained by the approximation method, the dashed black line marks the best test error obtained by the classical linear solver, and the grey line is the best test error obtained by the conjugate-gradient (CG) method. 4.16 gives the speed-up factors obtained by the Toeplitz-based RLS method with respect to the comparison strategies.

As predicted by theory, the approximation method proved very effective whenever the optimal parameter setting, σ^C , lead to a matrix that was close to a Toeplitz matrix; this situation is exemplified by the Coverttype and Daimler datasets; in both cases, the approximated model almost matched the classical method.

The conventional linear-system solver and conjugate gradient required to allocate the entire kernel matrix, whose storage cost scaled as $O(n^2)$. This set severe limitations for those problems involving more than $2 \cdot 10^4$ patterns, as memory storage for the kernel matrix exceeded 3GB; memory occupation further increased when taking into account the overhead brought about by the linear system solver (e.g. Cholesky decomposition) and the dataset matrix. This made memory storage quite a demanding constraint even on powerful machinery. Conversely, the experiments showed that the approximated method required less than 300 MB of RAM, even for datasets including more than 10^5 patterns.

The Toeplitz-based approach made it possible to train classifiers also with large data sets including 50,000 and 100,000 patterns, as was the case for Coverttype; in those situations, empirical evidence showed that the best model obtained by the approximated method outperformed the "exact" solution obtained by classical methods (which had been trained on fewer patterns). In other cases, the approximated approach yielded a sub-optimal solution, although classification errors always kept within a reasonable range.

The (successful) event $\sigma^C \cong \sigma^T$ could be observed when the size of the training sets reached $1e3$ patterns or more. Whenever σ^C differed significantly from σ^T , the results on large-scale data sets did not match those attained by classical solutions (even with fewer patterns); conversely, whenever $\sigma^C \cong \sigma^T$, the approximation for large-scale learning proved effective. These considerations provide a designer with an operative criterion to verify the advantage of the approximation scheme in tackling large-scale problems or in supporting limited-resource implementations.

Speed-up values always proved very satisfactory, also considering that the classical Matlab Gaussian Elimination solver could benefit from a parallel implementation whereas the LTZ algorithm version was not parallelized. An important remark concerns the analysis of timing results, as the reported speed-up values only took into account the computational process involved by the solution of the linear system. Therefore, one might argue that, if one also considers the time required to work out the kernel matrix, speed-up values should be adjusted and would decrease accordingly. Actually, the total number of kernel evaluations is constant and scales exactly as $O(ln^2)$ (where l is the number of variables); therefore it does not compromise the advantage in complexity that is conveyed by the Toeplitz-based approximation. At the same time, modern technology approaches to kernel matrix computation make that process easily parallelizable [172]; on the contrary, parallelization of the optimization engine is a difficult task. The experimental results presented in 4.15 confirmed the expected behaviors in terms of computational complexity as per `toeplitzTable1`. The speed-up provided by the Toeplitz solver over the conjugate-gradient method was roughly proportional to the number of iterations of the CG algorithm. Actually, in half of the experiments

the CG didn't attain the required solution accuracy before reaching the maximum number of iterations. Thus, the average speed-up provided by the proposed strategy is somewhat underestimated.

A comparative analysis with iterative methods for SVM [173] showed that the computational performances attained by the latter ones were heavily affected by the specific values of the regularizing parameter, λ (C SVM parameter in that case), whereas the performances of the Toeplitz-based method are guaranteed to keep constant and independent of that quantity. Likewise, the CG was heavily influenced by both parameters λ and σ : the computational complexity of CG increases when σ takes on high values and λ takes on low values. At the same time, empirical evidence pointed out that the accuracy provided by the approximated method matched satisfactorily the accuracy attained by iterative methods (CG), especially when large data set were involved.

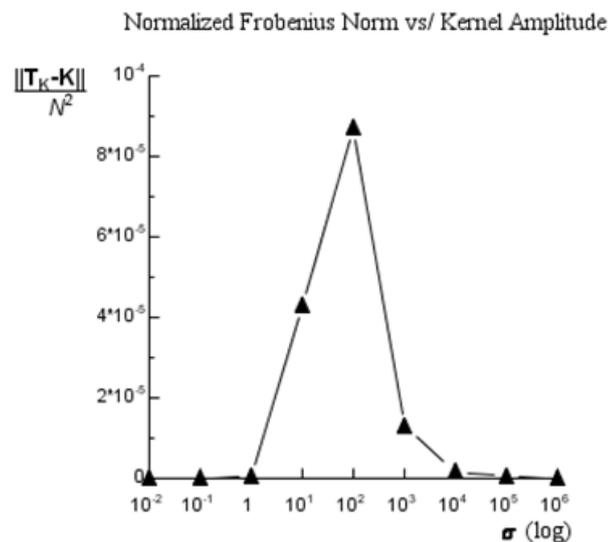
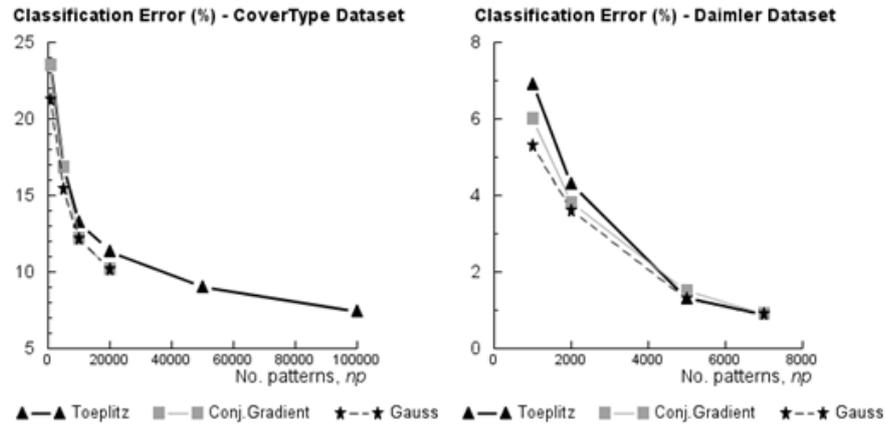


Figure 4.14: Normalized Frobenius norm of the error matrix versus kernel amplitude.

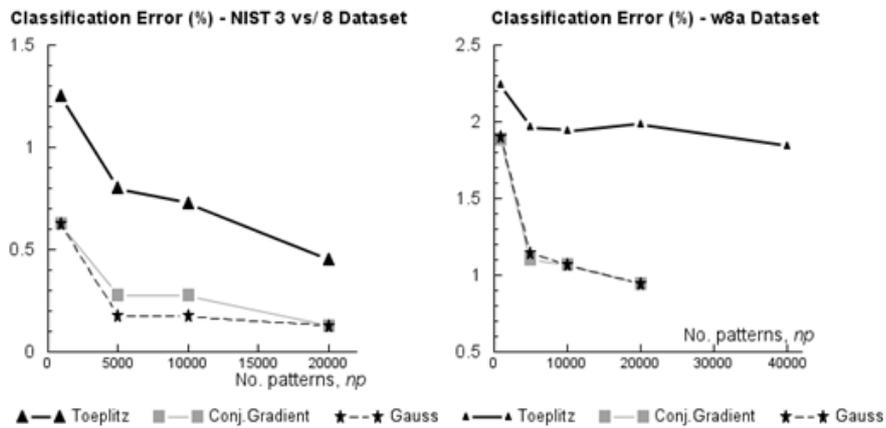
4.3.2.2 A Practical Procedure to Validate the Approximation for Large Datasets

The theoretical analysis and the experimental verifications point out that the use of an RBF kernel most likely enhances the Toeplitz-based scheme in



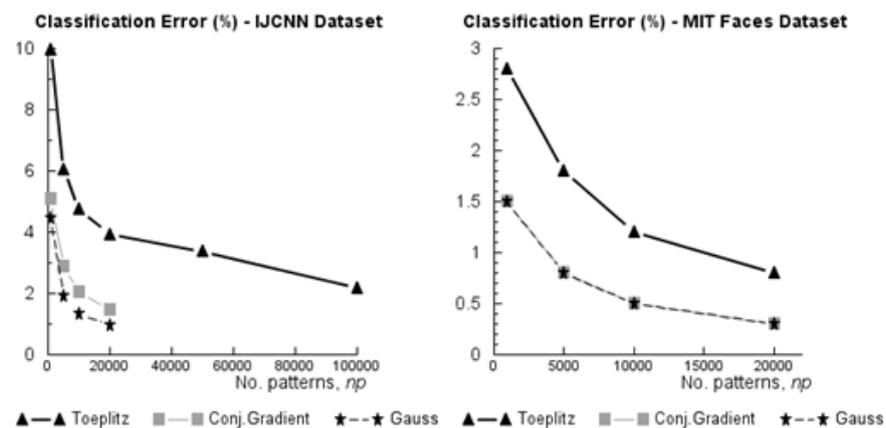
(a)

(b)



(c)

(d)



(e)

(f)

Figure 4.15: Accuracy comparison for Gaussian Elimination, Conjugate Gradient and Toeplitz Approximation

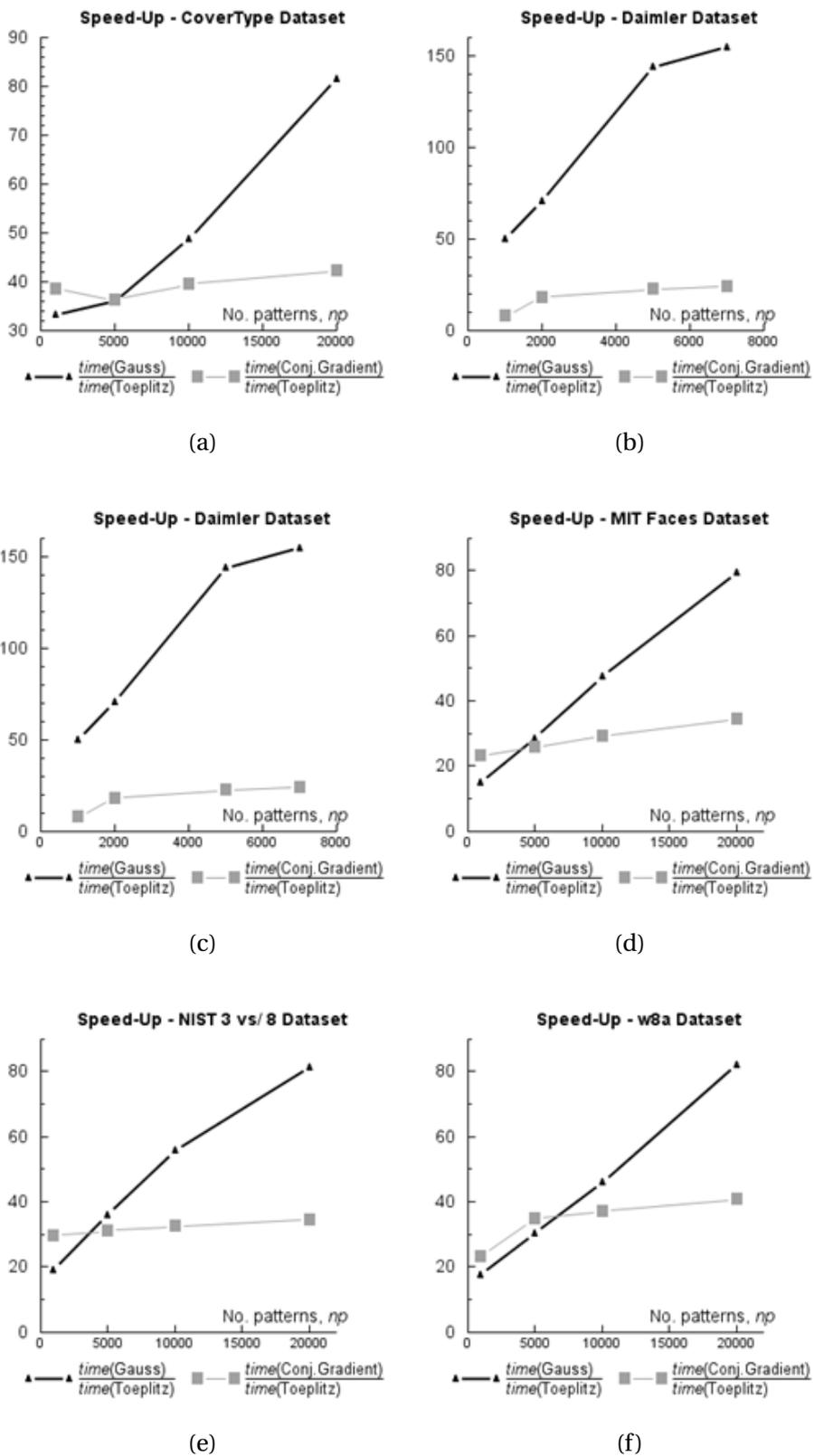


Figure 4.16: Experimental results for the RLS Toeplitz-based acceleration.

RLS learning tasks. In practical applications, one is interested in the accuracy of the eventual system set-up, and the effectiveness of the Toeplitz-based approach ultimately depends on the consistency of the model-selection process. In other words, one would know in advance whether the approximated schema yields classification results that are close to the solution attained by direct methods.

The above discussion and the analysis of empirical evidence provides an operational, reliable procedure to assess the validity of the approximation in a practical fashion, and is especially useful in large-scale training problems.

The method baseline is the verification that, whenever the approximated model parameter proves close to the 'correct' value ($\sigma^T \cong \sigma^C$) for a relatively small training set, then this property also holds for the larger data set; thus one has $T_K \cong K$ and the Toeplitz-based approach dramatically simplifies the training task on the actual domain. The operational procedure is therefore the following.

1. Input: a large set, \mathbf{X} , of training patterns;
2. Subsample \mathbf{X} and assemble a reduced training set, \mathbf{X}' , holding a number of patterns, n' , that can be managed by direct-solution methods (e.g. $n' \propto 1e3$);
3. Perform a model selection process on \mathbf{X}' by using both the direct-solution and the Toeplitz-based method, yielding model parameters $(\sigma^C)'$ and $(\sigma^T)'$, respectively;
4. if model selection is consistent, i.e., $(\sigma^C)' \cong (\sigma^T)'$, train an RLS machine on the entire set, \mathbf{X} , by using the Toeplitz-based method; otherwise the approximated method might not apply.

The advantage induced by the low computational complexity of the approximation scheme makes the above procedure also valid when dealing with limited-resource devices, since in those circumstances one has to use an algorithm that can support the learning process on the target device.

4.3.3 Concluding Remarks

The work has analyzed the application of Toeplitz-related algorithms to the training problems involved by Regularized Least Squares. When considering the associate linear-system problem setting, the basic advantage of the proposed approach lies in the dramatic reduction in both computational complexity and storage requirements involved by a Toeplitz-based problem formulation.

The presented research has proved a sufficient condition that yields a Toeplitz kernel matrix for mono-dimensional problems, with the result of a very efficient learning algorithm leading to the exact solution. An approximated problem setting has been described for the general case of multivariate domains, where the role of RBF kernels has been analyzed.

The theoretical analysis and experimental results have shown that, whenever the actual model implies a kernel matrix that is close to the approximating Toeplitz matrix, the proposed approach attains a marginal degradation in accuracy and, by contrast, allows one to tackle large-scale problems that would have been otherwise inaccessible. The approximation scheme is also appealing for embedded implementations, involving for instance low-cost DSPs, that are strongly constrained in resources and that can tolerate sub-optimal accuracy performances.

4.4 Regularized Random Neural Networks

The research presented here addresses both theoretical properties and applicative extensions of the basic ELM model. The specific analysis tackles the effect of regularization methods on the ELM performances, and proposes operational responses to some crucial questions, namely, when regularization is actually required and which regularization method proves beneficial eventually.

From a theoretical perspective, novelty contributions lie in deriving the Vapnik-Chervonenkis dimension of the ELM analytically, and in proving the equivalence between the regularized approach and a conventional Regularized Least Squares algorithm [53] when the ELM kernel [174] is implemented. From an applicative viewpoint, the research compares two alternative strategies for regularization. The first strategy hard limits the eigenvalues of the hidden layer matrix, and Truncated SVD is the tool to implement that approach. The second strategy applies Tikhonov regularization (Euclidean norm of weights) to soft filter the eigenvalues. This study eventually shows that, first, regularization improves significantly generalization performance especially when limited training samples are available, and, secondly, that Tikhonov regularization always outperforms the TSVD-based regularization strategy. The latter outcome is of particular interest because it ultimately proves that a soft filtering of the hidden-layer matrix eigenvalues is the most effective regularization strategy. The experimental verification involved a variety of standard real-world benchmarks, including both classification and regression problems. Empirical evidence always confirmed the validity of the analytical framework.

4.4.1 Generalization ability of the basic ELM model

Regularized versions of the ELM learning model have been recently proposed and analyzed in the literature [175],[176],[177],[178],[174]. In [175],[177],[174] the ELM model is used as the kernel function in a Support Vector Machine (SVM); conversely, Tang et al. [176] apply Truncated Singular Value Decomposition (TSVD) to the ELM hidden-layer matrix. Those re-

search works attained interesting results; at the same time, the choice of the applied regularization method does not always seem to be justified.

The ELM model supports a wide class of activation functions [73]. The formulation does not allow any theoretical prediction of the generalization performances, and does not provide any control mechanism on the activation functions that are used in the learning stage. Statistical Learning Theory provides the analytical framework to address such cases, through the formulation based on the Vapnik Chervonenkis dimension d_{vc} , as shown by the following property.

Lemma 4.4.1. *The Vapnik-Chervonenkis dimension of an Extreme Learning Machine having N_h hidden neurons is: $d_{vc} = N_h + 1$.*

Proof. In the ELM model, the activation values of the hidden layer are computed by applying a fully unsupervised procedure, that does not take into any account the expected target values. As a consequence, the computation of \mathbf{H} can be regarded as a (fixed) preprocessing step not involved in the training process. The output layer of the network supports a straightforward Perceptron with N_h inputs, which processes the activation values mapped by \mathbf{H} . By applying the expression of the Vapnik-Chervonenkis dimension of a Perceptron [3] the assertion follows.

The proof of this result is straightforward, however an important aspect emerges: the randomness of the hidden layer is equivalent to an unsupervised pre-processing technique of the data; it has been already shown [77] that an unsupervised pre-processing of the data is able to dramatically reduce the d_{vc} of a learning scheme, in particular the K-Winner Machine model [77] adopts clustering to reduce complexity; instead here a more complex unsupervised mapping is used, that given by the random hidden layer.

Lemma 4.4.1 opens new vistas on the adoption of some controlling mechanism in the training procedure; in particular, the present research stems from established theoretical results [38], which bounded the generalization

ability of a neural network by controlling the norm of the output weights. One can adopt that approach by applying the Regularized Least Squares (RLS) method [53] to the ELM mapping, and obtain a new, augmented formulation of the basic ELM.

4.4.2 Augmenting ELM with a Regularization Term

RLS learning and ELM learning can be linked by considering RKHS properties. When applying the linear kernel on the activation values (2.128), one obtains the ELM kernel [174]:

$$\mathbf{K} = \mathbf{H}\mathbf{H}^t \quad (4.28)$$

Therefore the RLS cost function (2.80) can be rewritten as follows:

$$\min_{\beta} \|\mathbf{y} - \mathbf{H}\mathbf{H}^t\beta\|^2 + \lambda \beta^t\mathbf{H}\mathbf{H}^t\beta \quad (4.29)$$

If one identifies $\bar{\mathbf{w}} = \mathbf{H}^t\beta$ in (4.29) with the vector $\bar{\mathbf{w}}$ of ELM, one immediately gets:

$$\min_{\bar{\mathbf{w}}} \|\mathbf{y} - \mathbf{H}\bar{\mathbf{w}}\|^2 + \lambda \|\bar{\mathbf{w}}\|^2 \quad (4.30)$$

The cost expression (4.30) shows that when applying RLS optimization to the ELM mapping, one eventually obtains a Tikhonov functional that augments the original cost formulation by a regularization term. This proves that adding a regularization term to the basic ELM formalism is equivalent to applying a Regularized Least Square approach using the kernel $\mathbf{K} = \mathbf{H}\mathbf{H}^t$. As a degenerate case, ELM reverts back to the basic RLS when $\lambda = 0$. Likewise, using ELM with a linear activation function $g(\cdot)$ is equivalent to applying RLS with linear kernel $\mathbf{K} = \mathbf{X}\mathbf{X}^t$ and $\lambda = 0$.

Introducing a regularization mechanism typically improves numerical stability and yields notable generalization ability in the general case, hence one expects that the same control strategy should enhance ELM, too. The following Section introduces two different spectral regularization strategies for ELM, namely, Tikhonov regularization and Truncated Singular Value Decomposition (TSVD).

4.4.3 Regularization Strategies for ELM

The approach described in [46] connected the theory of regularized linear-system solutions to the learning problem, showing that 1) due to regularization, the training process operates as a filter on the singular values of \mathbf{H} (there the kernel was used instead), and 2) regularization contributes as a numerical stabilizer, thus favoring the treatment of critical, limited -sample problems.

Following that theoretical framework, the Tikhonov regularization approach can be interpreted as the introduction of a smoothness-based filtering function, which operates in the space of functions spanned by the non-linear activations of hidden neurons. Conversely, the cost formulation (2.129) can be augmented by a hard thresholding regularization strategy on the eigenvalues of the matrix \mathbf{H} . Truncated Singular Value Decomposition (TSVD) [46] provides a powerful tool to support that strategy. In the following the two different regularization strategies are presented.

4.4.3.1 Tikhonov Regularization for Extreme Learning Machines

The formulation (4.30) matches the formal setting of inverse problems if one replaces the data matrix, \mathbf{X} , with the kernel matrix, \mathbf{H} ; therefore, the problem can be tackled by Tikhonov regularization. Since \mathbf{H} plays the role of \mathbf{X} , the linear fitting is performed in the space mapped by \mathbf{H} .

A preliminary step is the minimization of (4.30), which is straightforward because it involves a convex functional cost. Nullifying the gradient of (4.30) with respect to $\bar{\mathbf{w}}$ leads to the solution:

$$\bar{\mathbf{w}}_\lambda = (\mathbf{H}^t \mathbf{H} + \lambda \mathbf{I})^{-1} \mathbf{H}^t \mathbf{y} \quad (4.31)$$

where \mathbf{I} is the identity matrix.

The Singular Value Decomposition of \mathbf{H} gives:

$$\mathbf{H} = \mathbf{U} \mathbf{S} \mathbf{V}^t \quad (4.32)$$

where \mathbf{U} is an $n \times n$ orthogonal matrix. The matrix \mathbf{S} is equal in size to \mathbf{H} ; its only non-null diagonal entries are the singular values, s_j . Finally, \mathbf{V} is an orthogonal matrix of size $N_h \times N_h$. It can be shown [46] that the filtering *device* supported by Tikhonov regularization can be written as:

$$F_\lambda^{TK}(S) \equiv \text{diag} \left(\frac{s_i}{s_i^2 + \lambda} \right) \quad (4.33)$$

Using the above definition, one obtains that (4.31) is equal to:

$$\bar{\mathbf{w}}_\lambda = V F_\lambda^{TK}(\mathbf{S}) \mathbf{U}^t \mathbf{y} \quad (4.34)$$

When $\lambda = 0$, the expression (4.34) reduces to the classical formula for pseudo-inversion computed by the SVD, and complies with the basic ELM formulation. When instead one has $\lambda > 0$, the regularization mechanism filters the singular values of \mathbf{H} and suppresses the insignificant entries ($s_j \ll \lambda$) while retaining the large singular values that appear relevant.

The presence of the regularizing term λ also induces a numerical stabilization of the critical inversion process to be tackled when solving (4.31). Theory shows that, in principle, ELM is a universal approximating model; in a complex problem, however, the classification task might require a large number of neurons, which might even exceed the number of samples. In such cases, computing the pseudo-inverse of \mathbf{H} may turn out to be critical from a numerical viewpoint. As a result, the ultimate effect of the regularizing term is the possibility to use a high number of neurons without affecting the numerical stability of the convergence process. Using a high number of neurons does not mean building an over-fitting network, since it is known that, to ensure a network's generalization ability, the norm of the weights is much more important than the number of hidden neurons [38].

The regularization strategy is useful in the presence of limited training samples, especially when prior smoothness-bound assumptions can improve the generalization performances of the resulting model. This behavior can be formalized in Bayesian terms: denote with $p(\bar{\mathbf{w}})$, $p(\bar{\mathbf{w}}|\mathbf{y})$, and $p(\mathbf{y}|\bar{\mathbf{w}})$ prior, likelihood and posterior probabilities, respectively; by applying Bayes law

one should maximize the posterior with respect to $\bar{\mathbf{w}}$:

$$\min_{\bar{\mathbf{w}}} -\log p(\bar{\mathbf{w}}|\mathbf{y}) = \min_{\bar{\mathbf{w}}} -\log p(\mathbf{y}|\bar{\mathbf{w}}) - \log p(\bar{\mathbf{w}}) \quad (4.35)$$

Assuming that $\bar{\mathbf{w}} \sim N(0, \sigma_{\bar{\mathbf{w}}}^2 \mathbf{I})$ and that $\mathbf{y} = \mathbf{H}\bar{\mathbf{w}} + \mathbf{e}$ with $\mathbf{e} \sim N(0, \sigma_e^2 \mathbf{I})$, then one has:

$$p(\bar{\mathbf{w}}) = \frac{1}{(2\pi\sigma_{\bar{\mathbf{w}}}^2)^{n/2}} \exp\left(-\frac{\|\bar{\mathbf{w}}\|^2}{2\sigma_{\bar{\mathbf{w}}}^2}\right) \quad (4.36)$$

$$p(\mathbf{y}|\bar{\mathbf{w}}) = \frac{1}{(2\pi\sigma_e^2)^{n/2}} \exp\left(-\frac{\|\mathbf{H}\bar{\mathbf{w}} - \mathbf{y}\|^2}{2\sigma_e^2}\right) \quad (4.37)$$

The chosen prior corresponds to giving a null vector for in the absence of any other information: this prior formulation is classical and seems the most reasonable approach when no prior knowledge is available. By substituting equations (4.37) and (4.36) in (4.35) one can transform the functional (4.35) in:

$$\min_{\bar{\mathbf{w}}} \|\mathbf{H}\bar{\mathbf{w}} - \mathbf{y}\|^2 + \frac{\sigma_e^2}{\sigma_{\bar{\mathbf{w}}}^2} \|\bar{\mathbf{w}}\|^2 \quad (4.38)$$

The formulation (4.38) is a special case of the general training problem (4.30), in which the regularization parameter is preset: $\lambda := \sigma_e^2 / \sigma_{\bar{\mathbf{w}}}^2$. This proves that the ELM model is maximizing the likelihood probability and not the posterior probability. The prior is consistent with Structural Risk Minimization and, intuitively, plays the role of the margin in Support Vector Machines.

In summary, the analysis has proved that the regularized ELM network (4.30) is equivalent to Tikhonov regularization when performed on the space induced by the random hidden layer (matrix \mathbf{H}); moreover, the functional cost (4.30) allows a direct Bayesian interpretation and ultimately reduces the model within the Structural Risk Minimization paradigm.

4.4.3.2 Truncated Singular Value Decomposition for the Extreme Learning Machine

Enhancing the cost formulation (2.129) by a regularization mechanism is equivalent to a filtering process on the eigenvalues of the matrix \mathbf{H} . Trun-

cated Singular Value Decomposition gives a direct mechanism to filter out the singular values of matrix, \mathbf{H} , hence this approach can provide an alternative regularization strategy with respect to that described in the previous Section.

The selection of the eigenvalues in the SVD is explicit and is accomplished by applying a threshold on the singular values: after computing the SVD (4.32), one nullifies the singular values in \mathbf{S} that are smaller than a threshold value, τ . If one denotes with $F_\tau^{TSVD}(\mathbf{S})$ the resulting diagonal matrix after such a filtering process, the filter function can be formally expressed as:

$$F_\tau^{TSVD}(\mathbf{S}) \equiv \begin{cases} 1/s_j & \text{if } s_j \geq \tau \\ 0 & \text{if } s_j < \tau \end{cases} \quad (4.39)$$

Thus the system defined by the reconstructed matrix $\hat{\mathbf{H}} = \mathbf{U}F_\lambda^{TSVD}(\mathbf{S})\mathbf{V}^t$ is solved by:

$$\bar{\mathbf{w}}_\lambda = \mathbf{V}F_\lambda^{TSVD}(\mathbf{S})\mathbf{U}^t\mathbf{y} \quad (4.40)$$

4.4.4 Experimental Results

This section considers the practical effects of the regularization strategies presented above. The aim of the experimental session was twofold: first, to analyze the effectiveness of regularization when the number of training patterns varies; secondly, to compare the effects on ELM of the Tikhonov regularization and the TSVD-based regularization. The tests involved both classification and regression problems with heterogeneous datasets; all input variables were normalized in $[-1, 1]$. To enhance the statistical robustness of test results, a model selection process always determined the number of neurons and the optimal regularization parameter. Each experiment was repeated 10 times ('runs') and followed a cross-validation approach. In each run, the empirical data set was randomly split into a training and a test set, and the weight values in the hidden layer of the ELM network were randomly re-generated; to ensure consistency, within each run, the same random settings for the hidden layer were maintained when comparing the two regularization alternatives. Thus the obtained results proved robust not only with

respect to the random data-splitting but also with respect to the randomness of the hidden layer.

The size of the hidden layer varied in the range $[5, 200]$, which was sampled at steps of 5 neurons. In the case of Tikhonov regularization, the control parameter λ spanned the interval $[10^{-9}, 1]$ by a tenfold logarithmic sampling. In the case of TSVD regularization, the range for τ was $[10^{-13}, 1]$.

Throughout the following, the graphs give the measured performances of the basic ELM model as compared with the Tikhonov and the TSVD regularization approaches. In each graphs, the x axis gives the number of training patterns, and the y axis gives the performance on the test set. Test performances were measured by either the error rate or the RMSE, for classification and regression problems, respectively.

4.4.4.1 Experimental Results on Classification Problems

The experimental session addressing classification problems involved a variety of common benchmarks, namely, Coverttype, Sonar, Ionosphere, Pima Indians Diabetes, and NIST handwritten Manuscript 3 vs/ 8. These datasets refer to the well-know UCI machine learning repository [100] and Libsvm repository [62].

Figure 4.17 presents the obtained results. Test performances confirmed that the regularization mechanism always improves on conventional, non-regularized solutions; another interesting aspect concerned the error variance, which sharply reduced when comparing TSVD/ Tikhonov versus ELM. Tikhonov regularization always outperformed Truncation of SVD.

The optimal number of neurons selected by the model selection procedure proved higher for the TSVD/Tikhonov enhanced models than those resulting from the basic ELM. This was consistent with theoretical expectation and confirmed the improved numerical stability induced by the regularization mechanism, which stabilized the solution and supported larger networks.

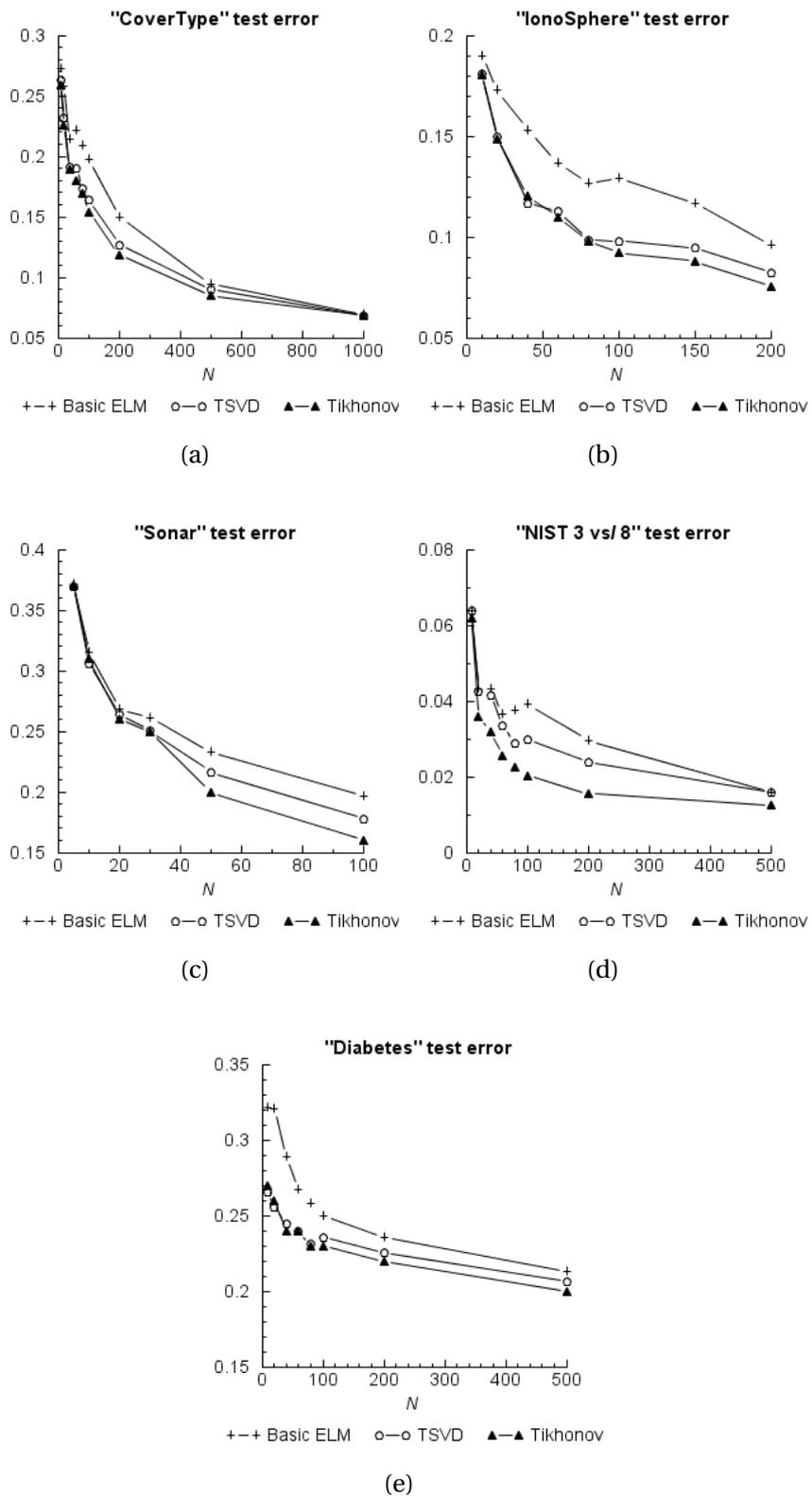


Figure 4.17: Classification error for ELM and Regularized ELM

4.4.4.2 Experimental Results on Regression Problems

The datasets used for regression problems were mostly drawn from the [179] repository: "Abalone", "Bank", "DeltaAleirons", "DeltaElevators", "Housing Boston", "Machine-CPU", "Stock", "Triazines", Wisconsin "Breast Cancer"; two additional datasets were drawn from the UCI Machine Learning repository [100], namely "MotorUPDRS" and "TotalUPDRS".

Figure 4.18 presents the results of this experimental session. Regularized networks always attained a neat improvement in accuracy with respect to the conventional EML model; at the same time, regularization lead to models involving a larger hidden layers. Alike classification tests, Tikhonov regularization always outperformed TSVD. Regression experiments highlighted a clear relation between the number of training patterns and the effectiveness of regularization; the beneficial effect of regularizing mechanisms proved more and more apparent when the number of patterns decreased, thus supporting theoretical expectations.

4.4.5 Conclusions

This work analyzed the learning scheme of ELM, and proved that ELM is equivalent to a Regularized Least Square approach when a particular kernel is chosen. This analogy supported the development of an operational framework, in which the generalization ability of ELM benefits from a regularization mechanism. This study therefore addressed the conditions and the features of spectral regularization strategies when applied to the Extreme Learning Machine model.

The regularization methods considered did not affect the computational efficiency of ELM training, whereas the filtering mechanism on singular values (induced by Tikhonov Regularization or by Truncated SVD) notably enhanced the generalization ability of ELM in both regression and classification problems. Furthermore, the research proved that a regularization strategy based on the soft filtering of the eigenvalues, such as the method formalized by Tikhonov regularization or the approach proposed in [174], is in general more effective than a strategy performing a coarse pruning of the singular

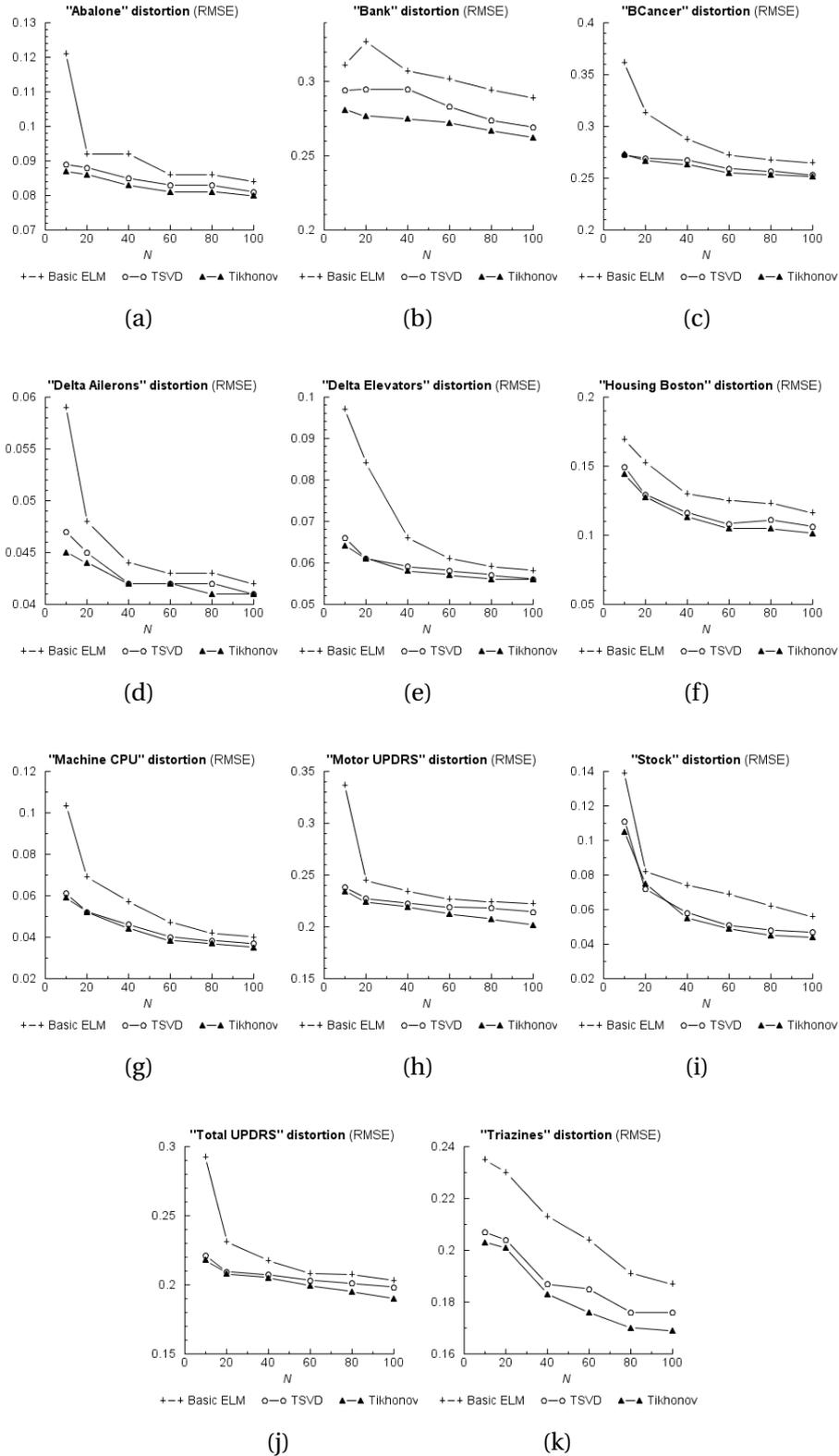


Figure 4.18: RMSE for Regression problems using ELM and Regularized ELM

values of \mathbf{H} . As a result, one can conclude that the most effective choice is to use the ELM kernel together with L_2 regularized kernel machines.

The resulting increased numerical stability allowed one to build, on average, larger networks. Empirical evidence also indicated a significant reduction in the variance of the prediction error/RMSE. Overall, the method performance proved less dependent on both the data sampling process and the randomness setting of the hidden-layer weights. Interestingly ELM induced kernels are less dependent for accuracy performance on regularization parameters than RLS with the usual kernels, for this reason selecting a proper regularization parameter is less critical than in kernel methods with classical kernels. Regularization was particularly effective in regression problems when the number of patterns is low, thus highlighting the interest of this model in critical, limited-sample situations.

4.5 Efficient Covariate Shift Detection by Clustering

In data-intensive applications, clustering based classifiers arrange huge amounts of data into a structured representation and search for relevant information [180][16]. The vast datasets and the heterogeneous descriptions of patterns set stringent requirements on the algorithms adopted; when empirical classifiers aim to optimize prediction on unseen data, attaining an accurate estimate of the run-time generalization error is a critical issue. Several methods in the literature have tackled that problem from both a practical [181] and a theoretical viewpoint [182][183][184].

Statistical Learning Theory can yet be of practical significance when dealing with clustering and data-mining [180], since data mining applications are typically rich in patterns and can therefore offer the required large samples. Moreover, the complexity of clustering-based classifiers often proves much lower than that of other approaches [180].

A crucial prerequisite in applying both theoretical and empirical predictions, however, is that the probability distribution of data is stationary [185]; such a condition holds in many practical testbeds and is often assumed to hold implicitly. In some data-intensive domains, however, the stationary nature of data distributions may prove questionable, either because the data refer to a phenomenon whose time-varying nature is overlooked, or because the original sample is so large that new samples stem from unexplored areas of the probability distribution, hence test data are virtually uncorrelated from training ones. Retraining (either from scratch or as an update of existing learning results) is a typical solution to that problem, but in data-intensive applications it may prove very expensive. This may occur for a variety of reasons: for instance, because the actual process for updating training results is difficult to design or implement, or because the amount of data is excessive, or because older data might not be easily accessible.

In general, non stationarity can be classified according to two different categories: *covariate shift* [186] refers to those cases in which the non stationarity only affects the pattern probability distribution $P(\mathbf{x})$; *concept drift* [187] refers to those case in which non stationarity is confined to the tar-

get probability distribution $P(c|\mathbf{x})$. This research addresses *covariate shift* and tackles the stationary-sampling issue from the conventional viewpoint of validation methods for classifier training [185]; to this aim, an efficient and reliable method to estimate the pdf and assess the stationarity of input data is required. To assess non stationarity the proposed criterion reformulates the original multivariate problem into an univariate problem by using a clustering procedure. This procedure avoids the curse of dimensionality and the possible numerical instabilities that can occur using traditional parametric methods such as Parzen Windows or Mixture of Gaussian Models [188][189][190][191].

The approach proposed in this work adopts the KWM model [77] as classifier. The rationale behind such choice is twofold. KWM yields tight bounds to generalization performance [77] and inherently supports multi-class classification tasks [192]. These features make KWM profitably suitable for data-intensive applications and for evaluating the applicability of Vapnik's generalization predictions accordingly, together with conventional cross-validation methods.

Experiments first show the approach validity in a synthetic domain, mainly to provide an intuitive demonstration of the basic non stationarity detection principle; the method is then tested in a group of complex real-world problems: the detection of intrusions in computer networks, Optical Character Recognition for numerical patterns, Emails Spam detection and Pedestrian Detection. The "KDD Cup 1999" dataset [193], the Manuscript NIST (MNIST) OCR dataset [99], the Spam Assassin dataset [194] and the Daimler dataset [195] provided the related experimental domains. Experimental results show that the proposed criterion successfully detects the non-stationary/stationary nature of the proposed domains.

4.5.1 Non Stationarity Detection for Assessing the applicability of generalization error estimation

Data-intensive applications pose the crucial issue of the stationary nature of the pattern distribution. In fact, the stationary-distribution assumption [196] is a basic prerequisite to the applicability Statistical Learning Theory.

Indeed, also empirical methods ultimately rely on the fact that the data distribution is consistently represented by the available sample [197], hence the assumption of a stationary distribution is critical in this case, as well. Non-stationary phenomena are in fact quite frequent in data mining, due to the time-varying nature of data or the huge size of the probability distribution that makes a complete sampling unfeasible. This in turn affects the reliability of the generalization error estimation associated to the trained classifier.

This research addresses this critical issue; a general criterion applies the clustering-based paradigm to evaluate the applicability of generalization prediction approaches when variations on $P(\mathbf{x})$ occur. The presence of a *covariate shift* on data [186] is a sufficient condition for the applicability of the developed method: from a cognitive viewpoint, this means that the pattern probability distribution, $P(\mathbf{x})$ is not stationary whereas the target probability distribution $P(c|\mathbf{x})$ satisfies the stationarity assumption. Such condition actually applies to most of the real world problems of practical interest. The proposed methodology, which is outlined in the following, in addition, can also successfully tackle problems in which non stationarity characterizes both the $P(\mathbf{x})$ and the $P(c|\mathbf{x})$ at the same time.

4.5.1.1 Non Stationarity Detection by Using Vector Quantization

In normal practice, one measures generalization performance by using a test set that is not involved in the training process. This is done for a variety of reasons: either because cross-validation drives model selection [181], or because the test set is partially labeled [3], or because the test set was not available at the time of training. Within that context, the assumption of a stationary distribution may be rephrased by asserting that, given a set $C = \{c^{(h)}, h = 1, \dots, N_c\}$ of N_c possible pattern classes, the training set instance, including n patterns, $T = \left\{ \left(\mathbf{x}_l^{(T)}, c_l \right), \mathbf{x}_l^{(T)} \in R^D, c_l \in C, l = 1, \dots, n \right\}$, and the test set instance, including n_u patterns, $S = \left\{ \left(\mathbf{x}_j^{(S)}, c_j \right), \mathbf{x}_j^{(S)} \in R^D, c_j \in C, j = 1, \dots, n_u \right\}$, are identically and independently drawn from a common probability distribution, $P(\mathbf{x})$. If such an assumption does not hold, the training set is not representative of the entire population classical estimates of generalization

can be unreliable.

The present analysis derives a general, yet practical criterion to verify the stationarity assumption, and consequently to validate the associate generalization error estimations. The proposed methodology tackles stationarity sampling from the conventional viewpoint of validation methods for classifier training [185]. The method uses a paradigm based on Vector Quantization (VQ) to check on the stationary-distribution assumption. A VQ-based classifier positions a set of prototypes so as to minimize some (unsupervised) distortion criterion in representing training data and calibration process observes the distribution pattern classes to assign a class to each prototype and the final step is the proposed criterion. The following conventions will be adopted:

1. $W' = \{(\mathbf{w}_n, c_n), \mathbf{w}_n \in R^D, c_n \in C, n = 1, \dots, n_h\}$ is a set of n_h labeled prototypes;
2. $\mathbf{w}^*(\mathbf{x}) = \arg \min_{\mathbf{w} \in W'} \{ \|\mathbf{x} - \mathbf{w}\|^2 \}$ is the prototype that represents a pattern, \mathbf{x} . The operator $\|\cdot\|^2$ here denotes the standard Euclidean distance.

Within the above conventions, the VQ-based stationarity criterion is outlined as follows:

1. First, one trains and calibrates a codebook, W , to classify training and test data.
2. Secondly, one estimates the discrete probability distributions, $P^{(T)}(\mathbf{x})$ and $P^{(S)}(\mathbf{x})$, of the training set, T , and of the test set, S , respectively; this is easily attained by counting the number of training/test patterns that lie within the data-space partition spanned by each prototype. Then the number of patterns of each cluster divided by the total number of patterns, constitutes the normalized frequency or 'bin' of the distribution.
3. Finally, one checks whether the data in S and T have been drawn from the same distribution.

Divergence	Notation	Function
Kullback-Liebler	$D_{KL}(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x}))$	$f\left(\frac{s_n}{t_n}\right) = \frac{s_n}{t_n} \log \frac{s_n}{t_n}$
Hellinger	$D_H(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x}))$	$f\left(\frac{s_n}{t_n}\right) = \left(\sqrt{\frac{s_n}{t_n}} - 1\right)^2$
Total Variation	$D_{TV}(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x}))$	$f\left(\frac{s_n}{t_n}\right) = \left \frac{s_n}{t_n} - 1\right $
Pearson (Chi-Square)	$D_P(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x}))$	$f\left(\frac{s_n}{t_n}\right) = \left(\frac{s_n}{t_n} - 1\right)^2$

Table 4.15: Theoretical formulation of divergence measures derived from the general class of f -divergences

In principle, several, different techniques may support the latter step. In the present approach and without loss of generality, $D(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x}))$ will denote a measure of divergence between the discrete probability distributions, $P^{(T)}(\mathbf{x})$ and $P^{(S)}(\mathbf{x})$. Any analytical measure of the discrepancy between two probability distributions is applicable for that purpose; in this regard, this work analyzes the performance of the general class of f -divergences [198][199].

Let be $f(t)$, a convex function defined for $t > 0$, with $f(1) = 0$. The f -divergence [199] of a distribution $P^{(S)}(\mathbf{x})$ from $P^{(T)}(\mathbf{x})$ is defined by:

$$D_f(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})) = \sum_{i=1}^{n_h} t_i f\left(\frac{s_i}{t_i}\right) \quad (4.41)$$

where s_i and t_i denote the normalized frequencies associated with $P^{(S)}(\mathbf{x})$ and $P^{(T)}(\mathbf{x})$, respectively. Different instances can be derived from the general class (4.41) by exploiting different implementations of the function f . Table 4.15 lists the most common divergences, which have also been adopted in this work.

The minimum (zero) value of $D_f(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x}))$ marks the ideal situation and indicates perfect coincidence between the training and test distributions. Non-null values, however, typically occur in common practice, and it may be difficult to interpret from such results the significance of the numerical discrepancies measured between the two distributions. The present research adopts an empirical approach to overcoming this issue by building

up a ‘reference’ experiment setting that constitutes the sample based threshold used to decide if a distribution is stationary or not.

The procedure can be outlined as it follows: first, one creates an artificial, stationary distribution, J , that joins training and test data: $J := T \cup S$. Secondly, one uses the discrete distribution J to draw at random a new training set, T_J , and a new test set, S_J , such that $T_J \cap S_J = \emptyset$. Both these sets have the same relative proportions as the original samples. Third, using these sets for a session of training and test yields a pair of discrete distributions, $P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x})$; finally, one measures the divergence between the new pair of data sets by computing $D_f \left(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x}) \right)$. This value provides the numerical reference threshold for assessing the significance of the actual discrepancy value $D_f \left(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x}) \right)$ by comparison.

If the original sample had been drawn from a non stationary distribution, then the associate discrepancy value, $D_f \left(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x}) \right)$ will be greater than the reference threshold $D_f \left(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x}) \right)$, computed on the artificial distribution J .

The following pseudo-code works out the complete procedure, which is characterized by low computational cost and high numerical reliability, compared with other estimation methods [189][191].

4.5.2 Discussion

For the sake of completeness and clarity, in the following the actions executed by the proposed procedure in two opposite scenarios are analyzed:

1. *Stationary Case.* If T and S are drawn from the same distribution $P(\mathbf{x})$, then T and S alone can be used to estimate the underlying $P(\mathbf{x})$. In other words, the discrete probability distributions $P^{(T)}(x)$ and $P^{(S)}(x)$ are both reliable estimates of $P(\mathbf{x})$ (under the assumption of a data-intensive problem). Thus, when applying step 5 of the proposed procedure (see pseudo-code above), one obtains a pair of sets (a training set, T_J , and a test set, S_J) that leads to a consistent estimates of $P(\mathbf{x})$ as well. As a consequence, $D_f \left(P_J^{(S)}(x), P_J^{(T)}(x) \right)$ must approximately coincide with $D_f \left(P^{(S)}(x), P^{(T)}(x) \right)$.

Algorithm 9 Criterion for validating the applicability of theoretical bounds

-
- 1: **procedure** VALIDATE(a training set including n_t labeled data, $\mathbf{x}_i, c(\mathbf{x}_i)$; a test set including n_s labeled data, $\mathbf{x}_j, c(\mathbf{x}_j)$)
 - 2: (Training) Apply a VQ algorithm on the training set and position the set of prototypes: $W' = \{(\mathbf{w}_n, c_n), \mathbf{w}_n \in R^D, c_n \in C, n = 1, \dots, N_h\}$
 - 3: (Probability distribution)
 - Estimate the training discrete probability distribution, \hat{T} as follows: $\hat{T} := \{P_n^{(T)}; n = 1, \dots, n_h\}$; where: $P_n^{(T)} = \{\mathbf{x}_i^{(T)} \in R^D : \mathbf{w}^*(\mathbf{x}_i^{(T)}) = \mathbf{w}_n\}$;
 - Estimate the test discrete probability distribution, \hat{S} as follows: $\hat{S} := \{P_n^{(S)}; n = 1, \dots, n_h\}$; where: $P_n^{(S)} = \{\mathbf{x}_i^{(S)} \in R^D : \mathbf{w}^*(\mathbf{x}_i^{(S)}) = \mathbf{w}_n\}$;
 - 4: (Estimating stationarity)
 - Compute normalized frequencies: $t_n = \frac{|P_n^{(T)}|}{n_t}$; $s_n = \frac{|P_n^{(S)}|}{n_s}$; $n = 1, \dots, n_h$
 - Compute the divergence between \hat{T} and \hat{S} : $D_f(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x}))$
 - 5: (Assessment of generalization error estimation)
 - 6: **if** a reference validation set is not available **then**
 - $J := T \cup S$.
 - Draw from J at random a training set, T_J , and a test set, S_J , having the same relative proportions of the original data sets;
 - 7: **else**
 - 8: Use the reference as S_J and set $T_J = T$
 - 9: **end if**
 - 10: Repeat steps (2,3,4) and then go to the next if
 - 11:
 - 12: **if** $D_f(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})) > D_f(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x}))$ **then**
 - 13: Data is stationary
 - 14: **else**
 - 15: Data is not stationary
 - 16: **end if**
-

2. *Non Stationary Case.* In this case T and S are drawn from two distinct distributions distribution $P_1(\mathbf{x})$ and $P_2(\mathbf{x})$ respectively. From T and S , one estimates $P^{(T)}(\mathbf{x})$ and $P^{(S)}(\mathbf{x})$ that are reliable estimates of $P_1(\mathbf{x})$ and $P_2(\mathbf{x})$ respectively; then, eventually, the reference value $D_f(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x}))$ is computed. If one merges-shuffles T with S , and split them with the same original proportions, as per step 5, one obtains the new pair of sets T_J and S_J . After working out the corresponding discrete probability distributions $P_J^{(S)}(\mathbf{x})$ and $P_J^{(T)}(\mathbf{x})$, the quantity $D_f(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x}))$ can finally be computed. This time, as the stationarity assumption does not hold, one expects $D_f(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x}))$ to be significantly larger than the reference value $D_f(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x}))$. Such discrepancy is caused by the fact that $P_J^{(S)}(\mathbf{x})$ and $P_J^{(T)}(\mathbf{x})$ cannot estimate consistently $P_1(\mathbf{x})$ and $P_2(\mathbf{x})$.

In the stationary case the merge-shuffle and split operations have no effect on distributions; instead, in the non stationary case, the merge-shuffle and split operations induce a change in the discrepancy values; $D_f(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x}))$ is always the adaptive threshold that allows one to discriminate between stationary and non-stationary distributions.

If stationarity is verified, the theoretical assumptions underlying Statistical Learning Theory hold, and the bound formulation or cross-validation are valid. Otherwise, when $D_f(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})) > D_f(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x}))$, one might infer that the original sampling process was not stationary, hence a direct application of theoretical results or empirical estimates is questionable. The artificial dataset (S_J, T_J) is built whenever a stationary ‘reference’ dataset is not provided. Indeed, the proposed procedure, when needed, the implementation of a resampling strategy, which can of course be repeated several times to enhance the statistical robustness of numerical estimates.

Two aspects make the methodology presented above suitable for data-intensive application:

1. In a single step one can obtain both the classification of data and the reliable estimation of the data distribution; this is due to clustering

2. the obtained estimation is based upon an univariate pdf, thus avoiding the usual issues that arise when classical methods such as Parzen Windows [189] and Mixture of Gaussians [191] when high dimensional spaces are involved.

4.5.3 Experimental Results

The aim of this section is to operatively investigate the previously proposed method. In particular a synthetic 2-D dataset is studied to intuitively show the effectiveness of the approach; further four real domains are analyzed. In this section all references to generalization bounds are assumed to be computed by using 2.146 in compliance with KWM theory. The parameter k present in 2.146 in all experiments is locked to 1: this simply means applying a 1-nearest-neighbor policy to the bound estimation, that in other words denote that one is not interested in a local estimate for each pattern but in a global one (see [77] for details).

4.5.3.1 Artificial 2-dimensional testbed

The experiments on an artificial testbed aimed at demonstrating the operational principles in a 2-D space that allowed visual inspection; in particular the following analysis, for simplicity, will only deal with the theoretical estimation method as per 2.146. The dataset simulated a context in which the test distribution progressively diverged from the original training one. The overall experiment involved 5 sets of data as per 4.19: the basic pair included the original training set (X_{tg}) and an associate test set (X_{ts}), which was drawn from the same distribution, thus mimicking a stationary case. Three additional samples (X_{ts1} , X_{ts2} and X_{ts3}) emulated non-stationary cases.

All datasets involved binary classification problems; the non-stationary phenomenon was emulated by generating data from a Normal distribution whose mean value drifted progressively. Thus the three sets X_{ts1} , X_{ts2} and X_{ts3} , could be interpreted as time-dependent variation laws of data distributions. This artificial experiment clearly did not require any re-sampling strategy to get a stationary reference.

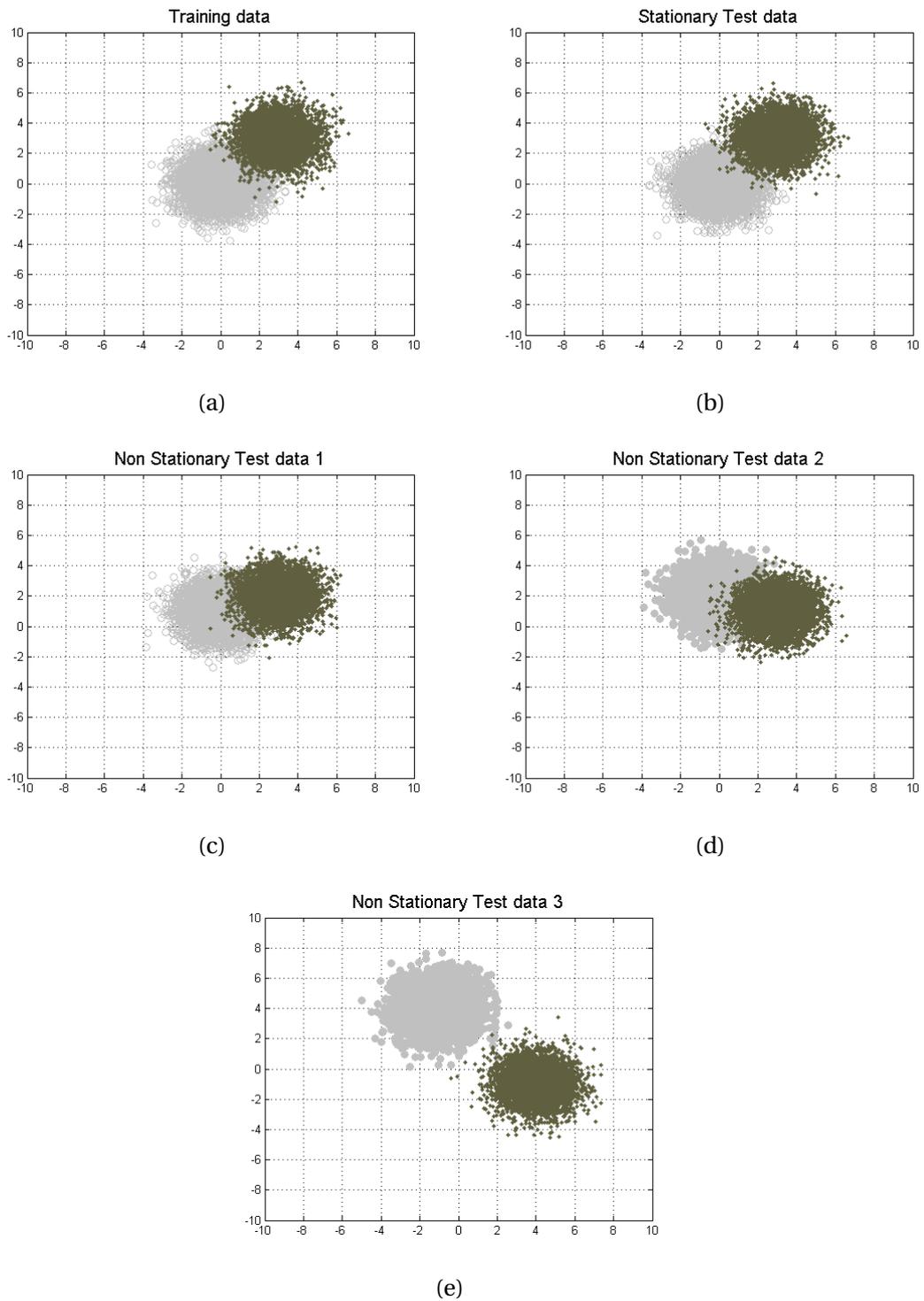


Figure 4.19: 2-D artificial dataset with a non-stationary data distribution. a) Training Set, X_{tg} b) Stationary Test Set, X_{ts} c) Test set, X_{ts1} d) Test set, X_{ts2} . e) Test set, X_{ts3}

n_h	Xts	$Xts1$	$Xts2$	$Xts3$
10	0.000852	0.304446	0.98017	2.847543
20	0.001781	0.372826	1.23305	2.582903
50	0.004367	0.43368	1.32496	2.417632
100	0.009302	0.460772	1.34089	1.573968
200	0.017330	0.441207	1.22429	1.528121

Table 4.16: Kullback-Liebler divergence D_{KL}

For each pair of sample, a range of clustering settings were tested, by increasing the number of prototypes and measuring the performances of the resulting KWMs. The discrepancy values associated with clustering outcomes progressively reflected the non-stationary nature of the phenomenon; at the same time, the error bounds predicted by generalization theory became more and more unreliable. Tables 4.16, 4.17, 4.18, and 4.19 give the values of the implemented discrepancies obtained from the analysis on the test sets. The values, computed as illustrated in Table 4.15, progressively increased from the stationary data set, Xts , up to the most ‘distant’ data set, $Xts3$. The relevant property in these results is that all measurements, albeit derived from different formulations, exhibit a common trend, thus supporting the method validity and robustness. Table 4.20 compares the error bounds predicted by generation theory for different numbers of prototypes with the actual errors measured on the various test sets. Empirical evidence confirmed that the bound values became more and more inaccurate and followed the same progression marked by the discrepancy values.

The graphs in 4.20 summarize the obtained results and clarify the divergence-based criterion in an intuitive way; in each graph, the x axis marks the different test sets, whereas the y axis gives each divergence formulation. The curves are plotted for various settings of the number, n_h , of VQ prototypes, and always witness a sharp increasing trend in discrepancy as long as the non-stationary test distribution diverges from the training sample.

Likewise, the graph in 4.21 demonstrates that the validity of the theoretical error bound progressively weakens in the presence of increasingly non-

n_h	Xts	$Xts1$	$Xts2$	$Xts3$
10	0.000428	0.136163	0.3556	0.677328
20	0.000897	0.170551	0.47083	0.806017
50	0.002196	0.194748	0.53416	0.981338
100	0.004583	0.204389	0.5598	0.945832
200	0.008438	0.203124	0.54957	0.933226

Table 4.17: Hellinger divergence D_H

n_h	Xts	$Xts1$	$Xts2$	$Xts3$
10	0.0298	0.6068	1.0086	1.2560
20	0.0414	0.6634	1.0820	1.4142
50	0.0718	0.7034	1.1546	1.4696
100	0.1048	0.7216	1.1858	1.3340
200	0.1370	0.7183	1.1605	1.2638

Table 4.18: Total Variation D_{TV}

n_h	Xts	$Xts1$	$Xts2$	$Xts3$
10	0.00172	0.50750	1.21620	2.22382
20	0.00368	0.77494	2.58869	4.72398
50	0.00901	0.94265	4.53441	17.34433
100	0.01789	0.99850	5.93358	43.91379
200	0.03280	1.04357	6.65349	95.72581

Table 4.19: Pearson (Chi-Square) D_P

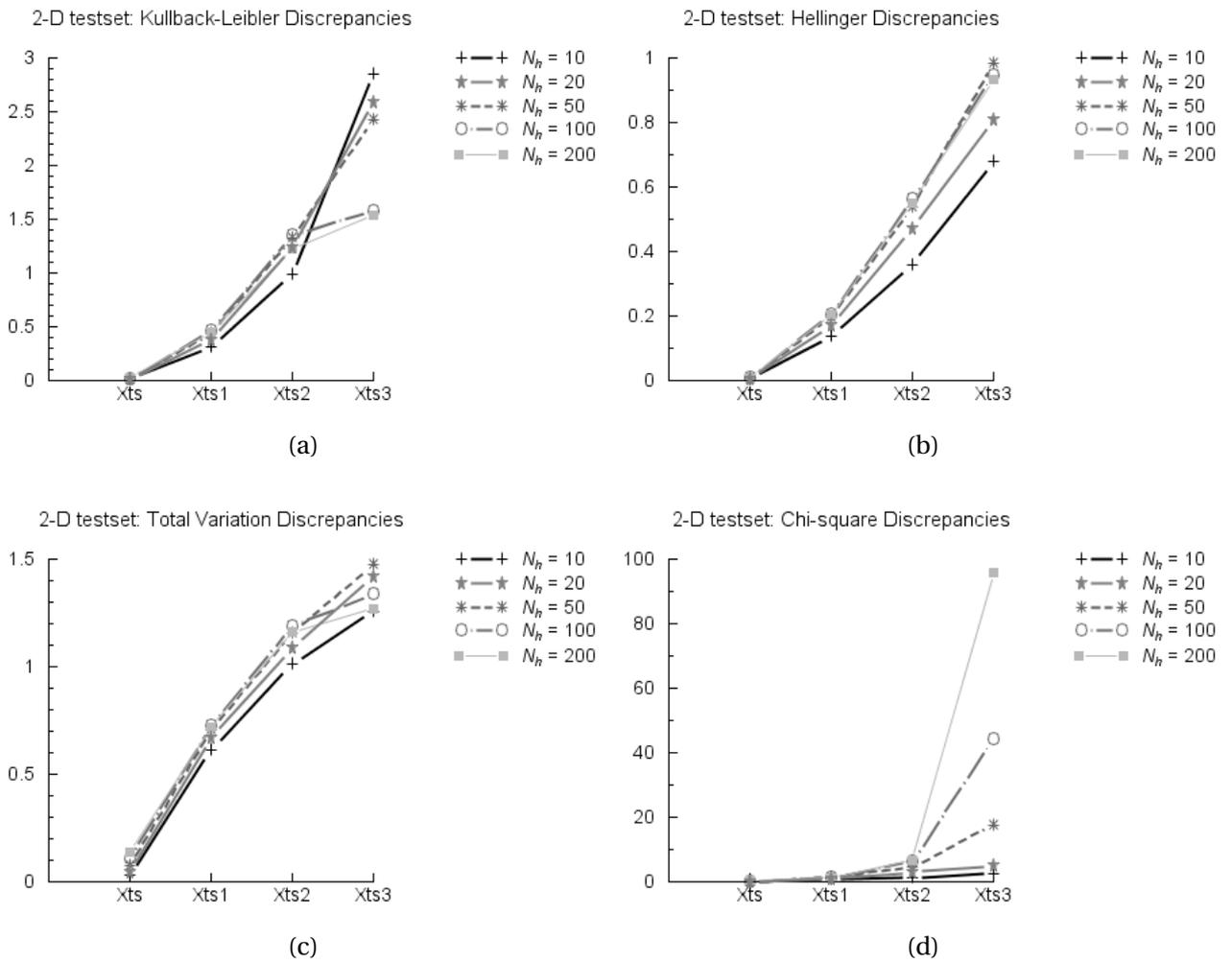


Figure 4.20: Discrepancy measurements for the 2-D artificial experiment. The curves are parameterized by the number of prototypes, n_h . Discrepancy values increase in the presence of non-stationary distributions of data.

n_h	R_{emp}	Bound	R_{Xts}	R_{Xts1}	R_{Xts2}	R_{Xts3}
10	2.78%	3.93%	2.59%	12.20%	33.40%	54.93%
20	2.05%	3.52%	2.16%	7.00%	18.23%	46.11%
50	1.89%	4.41%	2.18%	7.95%	21.90%	41.20%
100	1.81%	5.89%	1.85%	7.53%	21.10%	46.31%
200	1.67%	8.62%	1.82%	8.44%	25.80%	54.49%

Table 4.20: Training error R_{emp} , bound, and actual classifications error for the artificial dataset pairs

stationary distributions. Indeed, the predicted error bound from Statistical Learning Theory holds for the stationary case (Xts) only, as expected from the analysis presented above. As long as a non-stationary distribution takes place, discrepancy values increase and generalization performances diverge progressively from the theoretical bound.

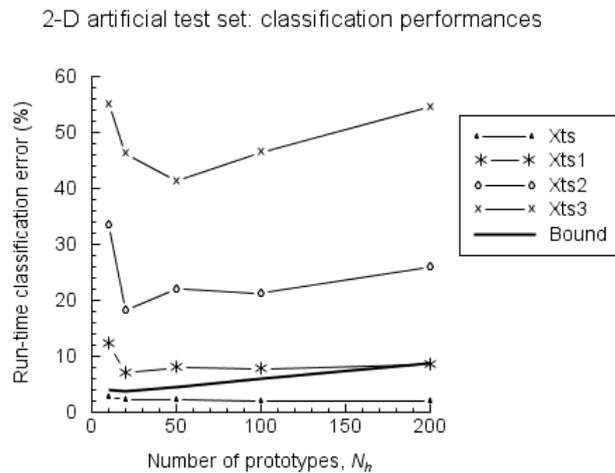


Figure 4.21: Generalization bounds and true classification performances. Theoretical predictions may prove unreliable by the presence of non-stationary distributions; Xts is the only case involving a stationary distribution.

4.5.3.2 Intrusion detection in computer networks: the “KDD Cup 1999” dataset

The data set used for the network-intrusion testbed was originally created for the Third International Knowledge Discovery and Data Mining Tools Competition [193]. The KDD dataset [193] originated from the 1998 DARPA Intrusion Detection Evaluation Program managed by MIT Lincoln Labs [200], with the objective of surveying and evaluating research in intrusion detection.

The original data spanned a 41-dimensional feature space; crucial descriptors that took on categorical values, most notably “Protocol_type” and “Flag”, were remapped into a mutually exclusive numerical representation, thereby leading a 52-dimensional feature vector.

Each pattern encompassed cumulative information about a connection session. In addition to “normal” traffic, attacks belonged to four main macro-classes. The complete training set contained about $5 \cdot 10^6$ patterns; normal traffic represented about 20% of the whole dataset, while attack types were quite unbalanced, as just two classes (‘neptune’ and ‘smurf’) spanned 78% of the entire dataset. The experimental session in this research involved a smaller training set, provided by the KDDCup’99 benchmark, which had been obtained by subsampling original training data at a 10% rate. The resulting “10% training set”, T , included 494,021 patterns and preserved the original proportions among the five basic categories. The test set, S , provided by the KDD challenge held 311,029 patterns, and featured ‘novel’ attack schemes that were not covered by the training set.

To verify the stationary nature of the observed data distribution, the procedure described in previous section compares the original distribution (T,S) with the representation supported by the exhaustive distribution, $J = T \cup S$, that approximated a stationary situation. The artificial, reference training and test sets, T_J and S_J , were obtained by randomly resampling J . The measurement of the various divergences between the training and test coverages for both distributions (T,S) and (T_J, S_J) completed the validation process. Table 4.21 gives the empirical results obtained for increasing codebook sizes.

n_h	Distribution	D_{KL}	D_H	D_{TV}	D_P
250	(T, S)	0.988	0.3	0.6910379	88.868469
277	(T, S)	0.997	0.301	0.6970229	94.806881
292	(T, S)	0.961	0.305	0.6987374	44.565382
294	(T, S)	1	0.303	0.6999451	82.754549
295	(T, S)	1.03	0.303	0.6969399	62.729348
367	(T_J, S_J)	0.0026	0.001299	0.0399745	0.0055132
369	(T_J, S_J)	0.00297	0.00147	0.0421336	0.0061434
370	(T_J, S_J)	0.0029	0.001414	0.0403167	0.0060123
371	(T_J, S_J)	0.00296	0.001448	0.0426404	0.0061934
380	(T_J, S_J)	0.0032	0.00155	0.0442762	0.0066035
388	(T_J, S_J)	0.003	0.00141	0.0423159	0.0059061

Table 4.21: KDD99: measured divergence values for the original distribution (T, S) and the stationary reference (T_J, S_J)

The number of prototypes, n_h , varied significantly in the two situations: when training and test data were drawn from a common distribution, J , the probability support was wider, hence the VQ algorithm required a larger number of prototypes to cover the data space. Conversely, the original training data, T , were drawn from a limited sector of the actual support region, thus a smaller codebook was sufficient to represent the sample distribution. Numerical results pointed out that the divergence for the original distributions (T, S) always turned out to be larger than the divergence measured when training and test data were drawn from a stationary distribution (T_J, S_J) . Such empirical evidence was mainly due to the marked discrepancies between training and test data sets, and clearly seemed to invalidate the applicability of the theoretical bounds from Statistical Learning Theory for the KDD99 dataset. In particular, this strong discrepancy, was characterized by the fact the real divergences are one or two order of magnitudes bigger than the reference ones. As a result, the validation criterion would predict that Vapnik's bound or cross validation error, would not hold for the original challenge data. For the sake of completeness, Table 4.22 compares the actual classi-

n_h	Distribution	R_{emp}	R	Bound
251	(T, S)	0.58%	7.83%	1.21%
278	(T, S)	0.36%	7.81%	0.95%
293	(T, S)	0.54%	6.66%	1.23%
295	(T, S)	0.37%	7.91%	0.99%
296	(T, S)	0.56%	8.00%	1.26%
367	(T_J, S_J)	2.32%	2.61%	3.65%
369	(T_J, S_J)	2.01%	2.36%	3.27%
370	(T_J, S_J)	2.22%	2.58%	3.53%
371	(T_J, S_J)	2.17%	2.52%	3.47%
380	(T_J, S_J)	2.03%	2.42%	3.32%
389	(T_J, S_J)	2.17%	2.51%	3.51%

Table 4.22: KDD99: training error R_{emp} , actual error R , bound, for (T, S) and the stationary reference (T_J, S_J)

fication errors with the theoretical bounds for the original and the stationary distributions.

These results are also given in a graphical fashion in 4.22 which also reports the cross-validation predictions of generalization.

Empirical evidence showed that theoretical or cross-validated predictions failed in bounding or predicting the generalization performance for the original data sets, whereas they provided good approximations as long as the sample distribution was artificially reduced to a stationary case. Such a conclusion gave both an empirical support and a numerical justification to a fact that has often been reported in the literature, namely, the notable discrepancy between the training and the test set in the KDD testbed. Such a critical issue had been hinted at by the proponents themselves of the competition [200], and explains the intrinsic difficulty of the challenge classification problem.

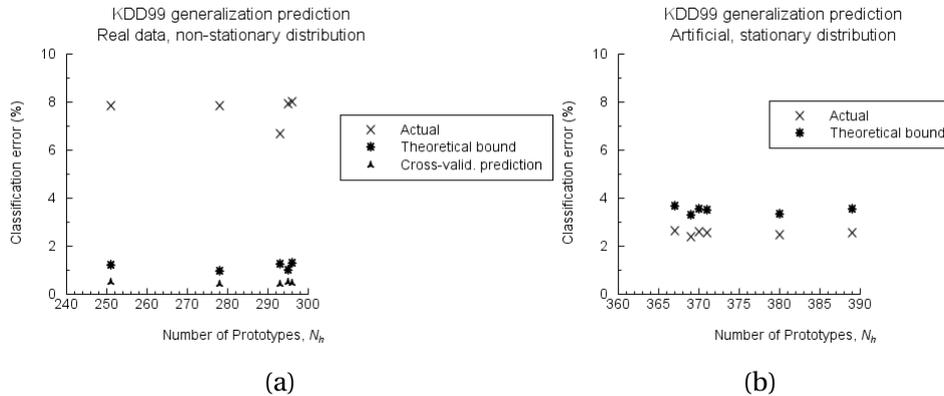


Figure 4.22: KDD dataset: validation of generalization predictions , a) original data, b) artificial stationary distribution

4.5.3.3 The Manuscript NIST dataset

The NIST handwritten digits database [99] provided an additional complex, real-world domain. This multiclass problem involves using three different data sets: a training set, X_{tg} , consisted of 60,000 samples, a test set, X_{ts} , and a validation set, X_{val} , including 60,000 and 58,646 samples, respectively. The data patterns underwent the same set of pre-processing steps that had been adopted and described in [201],[202]; the resulting set of features describing each character spanned a data space having dimension 80. As far as the proposed approach is concerned, the relevant fact of the MNIST data set is that the validation set, X_{val} , is quite uncorrelated with respect to the previous ones [201]; such critical issue, which was known in the literature, made it possible to verify the proposed bound-validating criterion in a real case with known and documented properties. When applying the validation procedure proposed, empirical results highlighted the marked discrepancy that characterized the pair of training and test data, (X_{tg}, X_{ts}) , with respect to the pair including the training and validation set, (X_{tg}, X_{val}) . Tables 4.23,4.24, 4.25,4.26 gives the experimental results obtained for the various divergence measures, for different settings of the codebook cardinality.

In all cases, divergence results clearly suggested that applying generalization theory or cross validation estimates to the validation set would yield

n_h	(X_{tg}, X_{ts})	(X_{tg}, X_{val})
50	0.000454174	0.093606572
100	0.001051512	0.14979364
150	0.00176112	0.191966419
200	0.002327155	0.223943361
250	0.003319328	0.236428103
300	0.003698878	0.252772341

Table 4.23: MNIST OCR domain, D_{KL} values

n_h	(X_{tg}, X_{ts})	(X_{tg}, X_{val})
50	0.0002271	0.0456244
100	0.0005248	0.0713274
150	0.0008796	0.0891992
200	0.001161	0.1048904
250	0.0016586	0.1096719
300	0.0018461	0.1180814

Table 4.24: MNIST OCR domain, D_H values

n_h	(X_{tg}, X_{ts})	(X_{tg}, X_{val})
50	0.0244	0.3375183
100	0.0365667	0.4301032
150	0.0459667	0.4692303
200	0.0533	0.5187436
250	0.0654667	0.5222731
300	0.0673	0.5428892

Table 4.25: MNIST OCR domain, D_{TV} values

n_h	(Xtg, Xts)	$(Xtg, Xval)$
50	0.000909213	0.192429682
100	0.002090747	0.296484197
150	0.003517104	0.367802702
200	0.004630067	0.460485661
250	0.006656246	0.489166082
300	0.007398064	0.551202225

Table 4.26: MNIST OCR domain, D_P values

unreliable bounds. Such a prediction was verified by testing the generalization performance of a KWM classifier (trained on the basic data set Xtg) on the MNIST validation set, $Xval$. Table 4.27 and 4.23 report on the obtained error measures, showing that empirical generalization errors always exceeded both worst-case theoretical predictions and cross-validation estimates on the original test set. An interesting result concerns the total variation obtained divergence values: on Tables 4.23 etc.. one can note that when the divergence value

$D_{TV}(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x}))$ is below $10 \cdot D_{TV}(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x}))$, then the corresponding actual error, is not so far from the theoretical bound. This observation empirically suggests that when $D_{TV}(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x}))$ is over $10 \cdot D_{TV}(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x}))$ the non stationarity level is significant, conversely when the divergence values is below that threshold then the associated test error will be not so far from the generalization error bound.

4.5.4 The Email Spam Database

This dataset involves a text-mining classification problem proposed in [203] and derived from the SpamAssassin Apache project [194]. Raw texts of emails have been mapped into a vector-space model for texts; a vocabulary of terms spans the feature space. The usual approach in building the vector space is counting the number of occurrences of each term [17] and then filling the data matrix with the corresponding term frequency for each text; conversely, in this case only the presence or absence of a term is recorded;

n_h	R_{emp}	Bound	Cross-Val prediction	(X_{tg}, X_{ts})	(X_{tg}, X_{val})
50	3.84%	6.00%	4.25%	3.79%	10.75%
100	3.01%	6.07%	3.12%	3.02%	9.96%
150	2.59%	6.45%	2.61%	2.60%	9.14%
200	2.44%	7.12%	2.32%	2.43%	8.46%
250	2.20%	7.61%	2.29%	2.19%	7.90%

Table 4.27: MNIST OCR predicted and empirical classification errors for the stationary pair (X_{tg}, X_{ts}) and the validation pair (X_{tg}, X_{val})

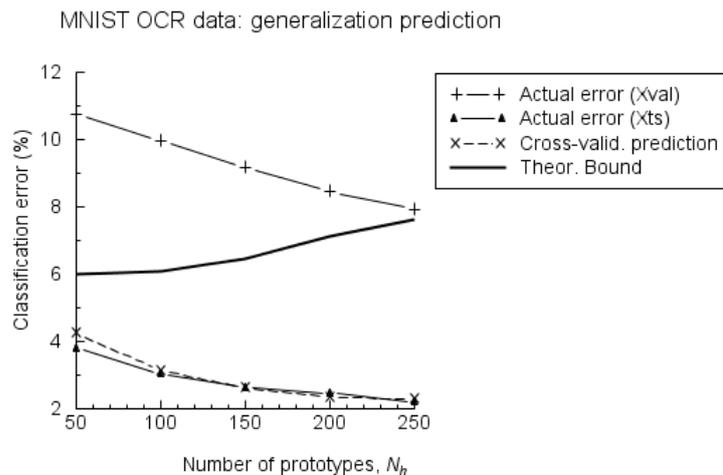


Figure 4.23: MNIST OCR domain: Predicted error performances and actual generalization performances for stationary and non-stationary test sets

this leads to a sparse data matrix composed only by '0's and '1's.

Emails are about 20% of spam and the remaining are legitimate emails traffic. Emails (and so patterns) are stored in chronological order so that one can capture the drift in time. The total number of emails is 9324 that were split in 2000 emails for training and 7324 emails for test: the split was performed such that the training set is composed by the first 2000 emails and the test set by the remaining 7324 emails, thus maintaining the chronological order.

This domain has a time varying distribution on input data: this is under-

n_h	Distribution	D_{KL}	D_H	D_{TV}	D_P
10	(T, S)	0.6857924	0.257748859	0.884818132	0.882226675
20	(T, S)	0.625897052	0.253369299	0.881237575	0.952071543
50	(T, S)	0.6583435	0.263773386	0.837421354	0.95870261
70	(T, S)	0.372328109	0.214588784	0.781716201	0.982382522
10	(T_J, S_J)	0.001462837	0.000733108	0.036357728	0.002974508
20	(T_J, S_J)	0.003157137	0.001554101	0.050715456	0.006035152
50	(T_J, S_J)	0.012887348	0.006095046	0.098995085	0.02310027
70	(T_J, S_J)	0.01575843	0.012957167	0.141682886	0.047113559

Table 4.28: Spam Assassin: divergence values for (T, S) and (T_J, S_J)

standable by considering that every time new spam arrives, correspondingly new words appear; this makes the data distribution non stationary in the input variable, e.g. the words of the emails and so consistent with the developed machinery.

Tables 4.28,4.29 give the obtained results in divergences, actual error on test data, and theoretical bounds. This dataset is strongly non stationary; also in this case the divergences values are much higher than corresponding references thresholds. Correspondingly the actual error on test original data is much higher than the predicted worst case bound. As per Manuscript NIST dataset a similar observation can be carried for the total variation divergence: under the empirical threshold of $10 \cdot D_{TV} \left(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x}) \right)$ the non stationarity makes the actual error not so far from the predicted bound.

4.5.4.1 The Daimler Pedestrian Detection Dataset

This dataset suggests an automotive application: the detection problem consists in discriminating between pedestrians against background objects. This testbed is composed of 9800 8-bits grey-scale images; 4 900 are pedestrian and 4900 are non-pedestrians. The dataset was split in 1225 training samples and 8575 test samples. Table 4.30,4.31 show the divergences values and the corresponding generalization error bounds. In this case the divergence values are almost identical to the reference threshold values; some divergence

n_h	Distribution	R_{emp}	R	Bound
10	(T, S)	12.85%	43.68%	19.49%
20	(T, S)	7.95%	32.38%	15.46%
50	(T, S)	5.40%	26.80%	16.88%
70	(T, S)	7.05%	27.13%	22.47%
10	(T_J, S_J)	11.3%	11.63%	17.61%
20	(T_J, S_J)	9.65%	10.3%	17.68%
50	(T_J, S_J)	7%	7.74%	19.26%
70	(T_J, S_J)	5.9%	6.64%	20.7%

Table 4.29: Spam Assassin: training classification error R_{emp} , actual error R , bound

n_h	Distribution	D_{KL}	D_H	D_{TV}	D_P
10	(T, S)	0.002415877	0.001211102	0.062040816	0.0048926
50	(T, S)	0.02385615	0.011747066	0.163965015	0.01275511
100	(T, S)	0.063620058	0.03018475	0.251195335	0.07802638
10	(T_J, S_J)	0.002681746	0.001354378	0.057609329	0.005606701
50	(T_J, S_J)	0.025458888	0.01252994	0.167696793	0.011140393
100	(T_J, S_J)	0.056902696	0.026487928	0.244198251	0.054021554

Table 4.30: Daimler: divergence values for (T, S) and (T_J, S_J)

values are higher than the reference ones, however these values are still much less than the empirical threshold $10 \cdot D_{TV} \left(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(x) \right)$. This suggests that the Daimler dataset is fully stationary; this conclusion is supported and confirmed by the associated generalization bounds that are never violated by the actual error on test data.

4.5.4.2 Summarizing comments

The obtained results exhibit interesting common features; among the various alternative formulations, Total Variation divergence appeared to be the most interesting. Its values are quite regular in all the experiments performed and

n_h	Distribution	R_{emp}	R	Bound
10	(T, S)	24.16%	22.67%	35.63%
50	(T, S)	15.34%	14.06%	37.08%
100	(T, S)	13.00%	14.53%	46.42%
10	(T_J, S_J)	19.26%	21.37%	29.74%
50	(T_J, S_J)	14.69%	14.90%	36.16%
100	(T_J, S_J)	14.11%	14.32%	48.15%

Table 4.31: Daimler: training classification error R_{emp} , actual error R , bound

allow one to decide an empirical threshold to distinguish among stationarity, modest non stationarity and severe non stationarity. In particular one can empirically assert the following rule of thumb:

1. $D_{TV}(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})) > 10 \cdot D_{TV}(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x}))$ marks a severe non stationarity and highly probable bound violations.
2. $D_{TV}(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x})) < D_{TV}(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})) < 10 \cdot D_{TV}(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x}))$ indicates a modest non stationarity and possible bound violations
3. $D_{TV}(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})) \leq D_{TV}(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x}))$ suggests the presence of full stationarity.

The last situation is less likely to occur because in every real world dataset exist a residual “physiological” non stationarity level. As in the Daimler case or Manuscript Nist (only the test set) case when

$D_{TV}(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})) \cong D_{TV}(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x}))$ then the dataset can be reliably defined stationary.

4.5.5 Conclusions

The baseline of the presented research is that non stationarity detection is a notable practical problem, especially in data mining problems where a huge amount of samples are provided. Generalization bounds from Statistical Learning Theory tend to become practical in the presence of large samples. At the same time, huge data sets drawn from complex distributions that

may be possibly time-varying or partially sampled pose the issue of the stationary nature of the observed data, which is a prerequisite for the reliability of generalization bounds.

The study has proposed a general and robust criterion for stationarity detection with consequent generalization error validation. The method exploits a clustering-based scheme for efficiently measuring the stationary nature of the observed data, and thereby assessing the consistency of generalization error estimation in data-mining applications.

The crucial aspect of the presented approach has been the empirical nature of the experiments in practical data mining; the cluster-based support of KWMs provided by Vector Quantization was used to build up a sample-based reference model and assess the stationary nature of the observed data accordingly. Indeed, in principle, the underlying model can be applied to any clustering-based classification scheme that prevents an uncontrolled increase in the d_{VC} . The specific analysis presented in this work was made possible by the tight bounds obtained when applying Vapnik's theory to the KWM model.

Intrusion detection in computer networks, manuscript numeral OCR, Pedestrian detection and Spam filtering have been adopted as case studies. In the first domain, the reference KDD99 data set case showed the validity of the criterion in a truly non-stationary, mission-critical context such as network-security systems. The second domain involving MNIST data made it possible to verify the criterion effectiveness in huge data sets stemming from incomplete sampling processes of complex distributions. The third dataset confirmed the effectiveness of the approach in another real world environment such as Text Mining and the last, Daimler, provided the stationary counter-example.

4.6 Underwater Port Protection by Machine Learning Tools

Target detection and intrusion prevention are crucial issues in undersea and port protection systems. The complexity of the involved problems and the lack of established models of the underlying phenomena have raised an increasing interest for adaptive paradigms such as Machine Learning and Neural Networks [204],[205],[206],[207].

In the specific scope of intrusion detection, neural classifier tools have already been successfully coupled with sonar-based systems [204],[205],[206],[207] to enhance accuracy in target detection [208]. Sonar technology is very effective in the monitoring of obstruction-free large volumes of water, but may fail in the presence of acoustically shadowed spots. This typically occurs in the proximity of the seafloor and is due to echo, reverberation, and others issues related to the morphology of the sea bottom [209]. Magnetic-based detection systems have proved effectively in those critical conditions where peripheral sensing is required [210]. Indeed, the current trend in the design of high-accuracy, high-reliability protection systems is to couple sonar-based and magnetic-sensing technologies, and to endow them with complementary missions [210]. In fact, the topology configuration and the operational deployment of a magnetic-based detection system is a critical issue due to: 1) the intrinsic, highly non-linear magnetic noise from the environment (e.g. a port, human activity, anomalies in solar activity), and 2) the peculiar nature of the weak signal sources associate with the targets of interest (e.g., divers). Moreover, an intrusion detection system should be also robust against random malfunctioning of the sensing devices.

Thus magnetic-sensing technologies need to be enhanced by the nonlinear representation and the classification ability of machine-learning models for accurate detection.

The novelty aspects of the presented research mainly lie in the integration of machine learning tools in a magnetic-based sensing architecture and in a design strategy to select the adequate paradigm in such a context. This methodology results in a highly accurate, highly reliable and robust detection system aimed at diver intrusion prevention and port protection. The

approach involves a wide variety of empirical tools, which range from unsupervised models for data representation and inspection, (Plastic Neural Gas [76], and Random Projections [80]), to supervised classifier models for target detection and identification (Support Vector Machines [3] and Circular Back Propagation Networks [66]).

The author apologizes for the lack of information about the on-field data collection process because at this moment this information is NATO classified.

4.6.1 Architectures for Magnetic-based Detection Systems

The adoption of the geomagnetic field for target detection allows one to manage critical scenarios that are not covered by other technologies (e.g., weak signal detection, protection of acoustically blind areas, etc). At the same time, one has to tackle a complex phenomenon, which results from the superposition of several magnetic fields generated by natural and artificial sources. Sources can be both internal and external to Earth, and are characterized by different physical qualities (i.e. form, position, etc...).

The classical detection approach highlights the magnetic characteristics of the target by separating them from the overall (geo)magnetic field. It requires an accurate modeling of the geomagnetic field, and assumes that one can single out and classify every magnetic source that generates interesting signals. In practice, an analyst typically tunes a bank of devices by adjusting the characteristic frequencies of Low-Pass, Band-Pass, and High-Pass Filters. The actual filter settings depend on: 1) the magnetic sources of information that are associated with the targets, and 2) the signals that are therefore predicted by the model. This method exhibits some crucial drawbacks and nowadays is regarded as ineffective, especially for the detection of weak signals [210]. This is mainly due to the complexity of the observed phenomenon which prevents the refinement of an accurate model of the measured field.

An alternative approach extracts the magnetic signal of the target by means of a flexible noise-cancellation technique, which does not make any assumption about the observed environment [210]. The critical operation of this method is an adaptive filtering procedure, in which one *measures* the back-

ground (noise) signal, and uses the empirical results to cancel the noise term from the total observed field; this ultimately highlights the information of interest. Such an approach does not imply any subjective setting of filters, but requires a careful sensing of the environmental noise.

The simplest adaptive set-up involves a twin-sensor configuration, in which one device captures the overall signal (also including target contributions), whereas the other measures the “reference” environmental magnetic noise. The proper displacement of the sensing devices clearly becomes a critical aspect to ensure a correct acquisition of the target-independent noise component [210].

Previous research [211] developed and tested the design guidelines for the proper topological configuration of the devices. When, ideally, both sensors are perfectly synchronized, a simple subtraction between the two measures highlights the target signal.

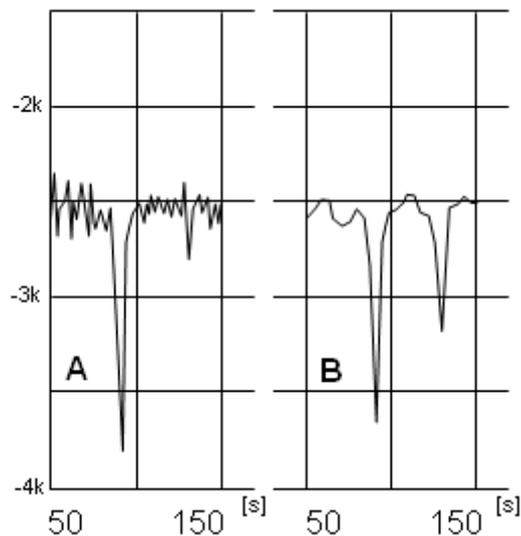


Figure 4.24: Comparison between the adaptive (A) and the classical (B) detection approach on a real diver-intrusion test. The classical filtering approach (involving the total magnetic field only) induces a false-positive.

4.6.1.1 Architectures for adaptive magnetic-based detection

The adaptive approach leads to two possible supporting architectures. In the basic schema, an array of *sentinel* magnetometers are all coupled with one separate device, which provides the *reference* measures of the noise signal to be canceled. This architecture is called RIMAN-type (Referred Integrated MAgnetic Network) configuration [211], and is shown in Fig 4.25 The Self-referred Integrated MAgnetic Network (SIMAN) schema improves on the basic RIMAN system [211]; it includes an array of sensors in which, at the same time, every instrument operates as a sentinel and as the reference for the neighboring magnetometers. Figure 4.26 shows the overall SIMAN architecture. The approach presented in this work implements a SIMAN configuration and includes, for the sake of simplicity and without loss of generality, a pair of magnetometers. In practice, at least three devices should be deployed in order to break symmetry and allow the detection of half-way crossing targets; nevertheless the twin-device experimental configuration is suitable for the validation of the system under a wide range of operational conditions.

4.6.1.2 Critical issues in the detection strategy

The integration of a reference and a sentinel signal greatly enhances detection effectiveness because it makes it possible to bypass the requirement of an accurate model of the observed environment. At the same time, the differential nature of the overall principle dictates strict design guidelines and poses some relevant issues.

First of all, the assumption of sensor synchronization is mostly unrealistic in real scenarios, which involve several magnetometers with non-ideal physical characteristics; misalignments in the time domain affect the detection system in the form of signal random perturbations. The latter terms add up to the inherent noise component that results from the imperfect coherence between the sentinel and the reference sources, due to the spatial displacement of the sensors.

Another critical aspect in the tuning of the pairwise architecture stems from the spurious signals that may be prompted by individual magnetome-

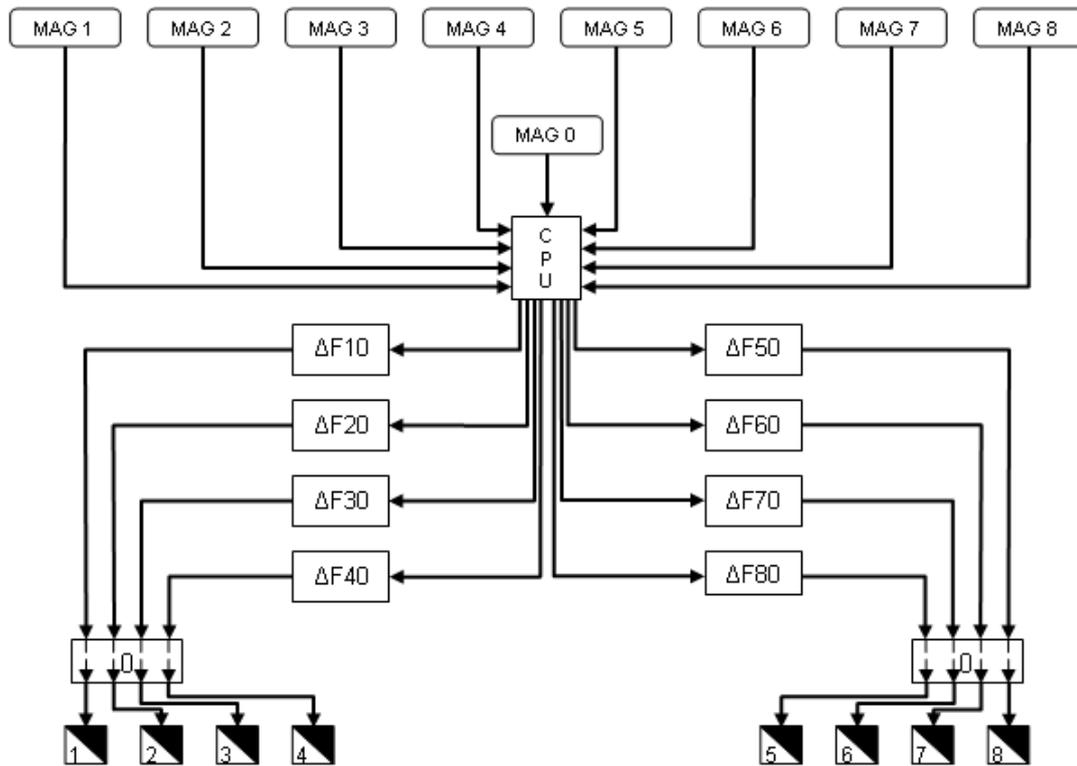


Figure 4.25: RIMAN adaptive configuration: the array of magnetometers are referred to one reference sensor (MAG0). Each blocks ΔF_{x0} supports the adaptive target detection of the x -th sensor.

ters during the sensing procedure. Spurious phenomena are only due to a defective behavior of the device; they appear in the form of extremely brief spikes and can mislead the detection system if they are not properly recognized as false alarms. These anomalous impulse signals affect most current magnetometers and are quite difficult to predict, since they are not related to the measurement procedure but only depend on the physical characteristic of the sensors.

The above variety of external critical issues exhibits a complex operational problem, whose main aspects are the highly nonlinear nature of the involved phenomena and the practical difficulty to predict the behavior of system components. The general crucial issue actually is the absence of any model of the overall scenario.

These considerations justify the adoption of empirical methods to sup-

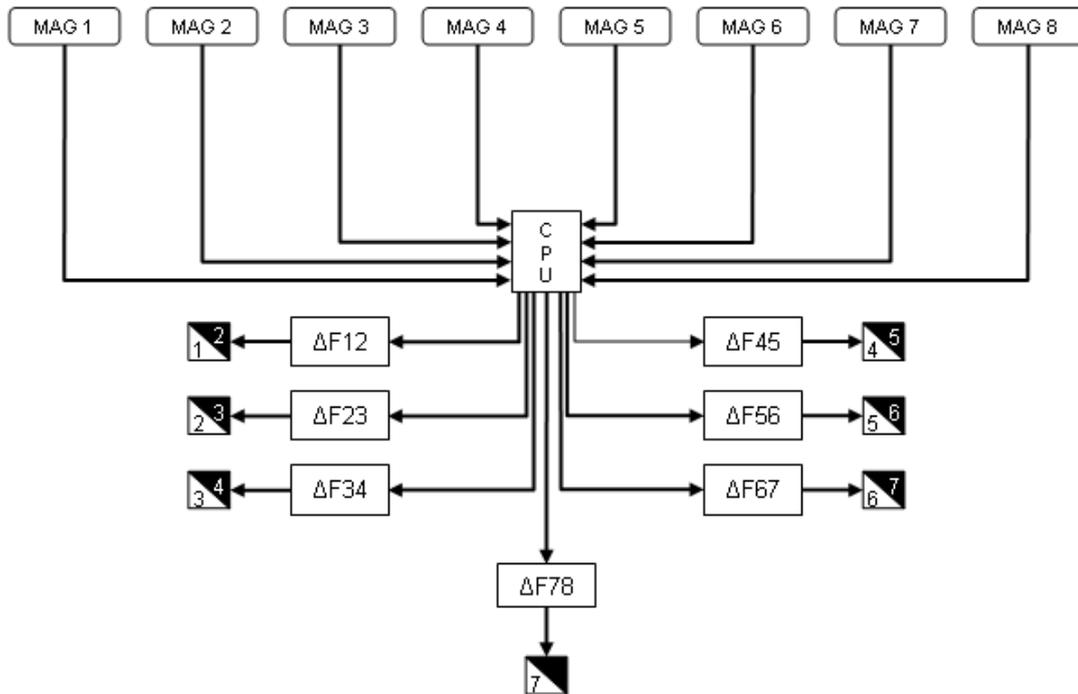


Figure 4.26: SIMAN configuration: each magnetometer operates both as a sentinel and as a reference for neighboring devices. Block ΔF_{xy} supports the adaptive target detection for the pair of sensors MAG_x and MAG_y .

port the detection system. The detection principle itself calls indeed for adaptive paradigms, which can adjust their performance in compliance with environmental conditions and specific signals. From this viewpoint, the field of computational Machine Learning (ML) techniques offers a wide spectrum of suitable methodologies, and the present research shows that ML-related paradigms can cover virtually the entire range of requirements for an effective deployment of the adaptive magnetic-based detection technologies.

4.6.2 Machine Learning Methods for Magnetic-based Detection

The main issues that are raised by magnetic-based detection technologies can be summarized in the following requirements: first, one needs adaptive methods to analyze raw data and extract the actual information content for the purpose at hand; secondly, specific and reliable paradigms should support the critical parts of the process, such as target detection and event clas-

sification; finally, one needs some quantitative method to validate the obtained results and compare the alternative solutions objectively. Machine learning techniques aim, on one hand, at developing data-analysis methods for feature selection and data visualization, and on the other at building predictive systems that can make reliable decisions on unseen samples. As a result, these paradigms can fit the magnetic-detection requirement quite effectively.

The ML methods used in this study selects the appropriate model for the magnetic signal detection problem from among three alternative methods paradigms: Plastic Neural Gas [76] provides the unsupervised clustering methodology; Support Vector Machines [3] and Circular Back-Propagation Networks (CBP) [66] are used as supervised models. In addition, random projections tools [80] are used to visually inspect data and validate classification results.

The research presented in this work adopts Random Projections (RP) techniques [80] for that purpose.

4.6.3 Overall Architecture of the Adaptive Detection System

This Section presents the architecture that integrates Machine Learning paradigms within the magnetic-based detection technologies, and illustrates the complete on-line processing chain that maps input signals into a decision result at run time. The data provided by the sensory system consist in a reference signal and in a target signal; to preserve as much as possible the original information content within data, both input streams are not either pre-filtered or processed in any way. This ensures that the result of the learning process in the adaptive neural component is not affected, hence no a-priori processing of signals might bias the inductive learning process.

Each input stream is segmented by applying a conventional windowing technique: the target and reference signal is arranged into frames including L samples; the resulting windows are allowed to overlap. The amount of overlap determines a tradeoff between sample size and information consistency: indeed, a small overlap reduces the number of patterns in a sample and miti-

gates the mutual correlation between time-consecutive input patterns; at the same time, too small an overlap extension might affect information consistency among loosely correlated patterns.

The research presented in this section aims at an objective evaluation of the overall method effectiveness, hence the preservation of the informative power of input signals is important. Thus the largest overlap ($L - 1$) between time-consecutive frames is adopted, and for an original stream having length h , the total number of patterns (windows) is $n = h - L + 1$.

As a result, the *input* space to the neural network components joins the stream segments from the reference and the sentinel sensors, and is therefore represented by a vector having dimensionality $m = 2L$. Since this strategy generates a huge amount of patterns, a random sub-sampling technique can reduce the final cardinality of the dataset for training the system. To facilitate the numerical convergence of the training algorithms, the generated dataset is normalized in the range $[-1, +1]$ for each time instant: in other words arranging samples data in a $n \times m$ matrix, normalization in $[-1, +1]$ is performed on each column.

The *output* information prompted by the neural detection systems adopts a hierarchical event-labeling schema: a preliminary classifier (Signal Classifier) module discriminates Normal Signal (NS) patterns from Anomalous Signal (AS); a subsequent targeted Event Classifier module processes the AS input streams and separates Target Signal events (TS, e.g., a diver intrusion) from Sensor Spiking signals (SS, i.e., an anomalous sensing malfunctioning in a magnetometer). Both classifier stages are supported by adaptive neural components described in the previous Section, and are trained empirically. The architecture described in Figure 4.27 presents some degrees of freedom in the choice of the neural technologies that actually support the various components. The adaptive nature of the design criteria allows one to complete that selection in compliance with the specific application context depending on empirical measures.

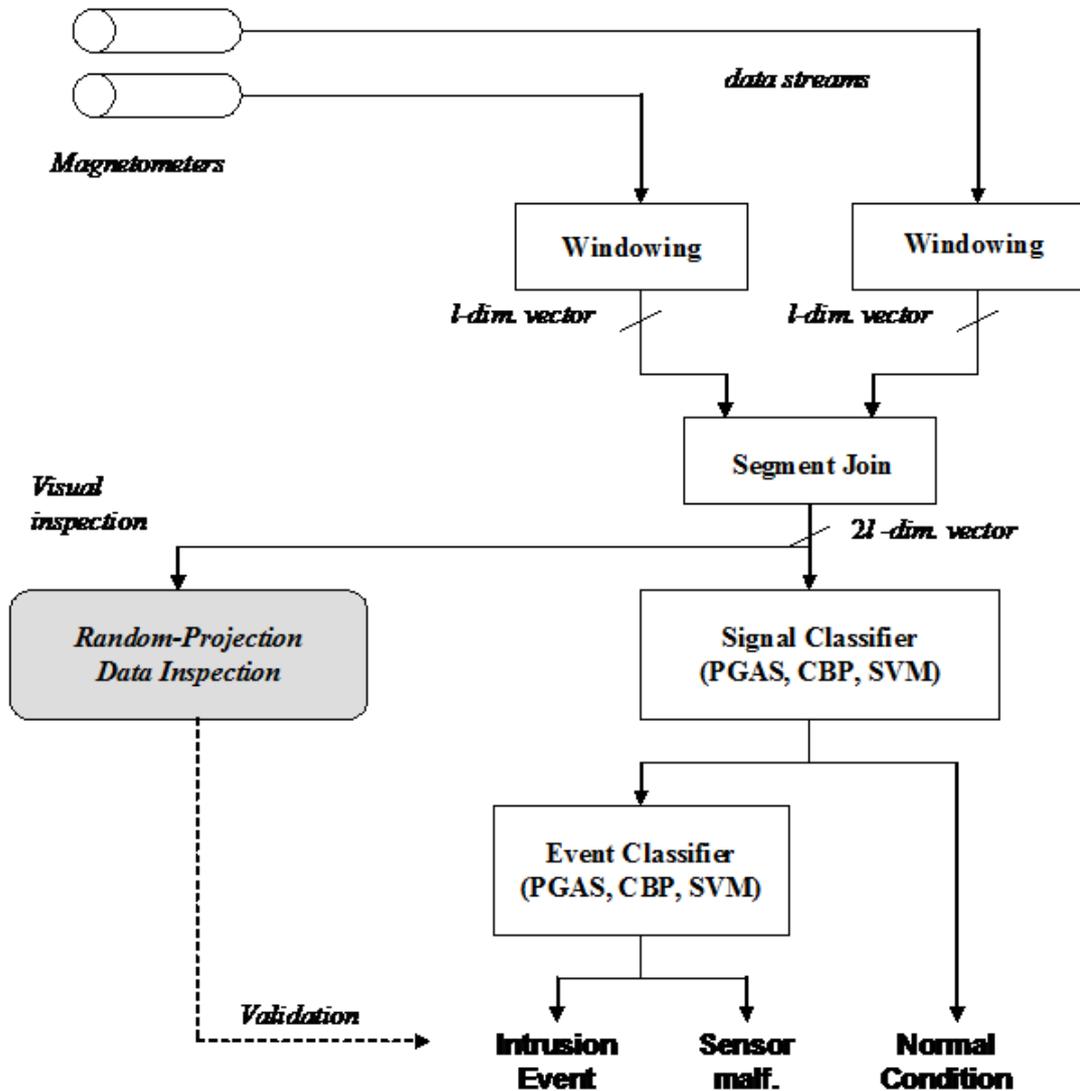


Figure 4.27: The processing architecture of the overall neural magnetic-based detection system.

4.6.4 Experimental Results

4.6.4.1 Experimental set-up

In the experimental set-up used for the verification of the magnetic-based detection approach, the sampling frequencies for the reference and the sentinel data streams were of 50 Hz. The detection response time, depended on the sampling frequency due to the window overlapping mechanism; after

Signal Classifier	Training set			Test set		
	Window size, L	-1	+1	Total	-1	+1
10	268	265	533	280	251	531
50	268	265	533	280	250	530
100	267	265	532	278	250	528
200	267	265	532	280	250	530

Table 4.32: Number of patterns for the Signal Classifier

transient completion, the system could prompt a detection response every 0.02s (i.e. 50 Hz sampling).

To attain a robust estimation of the generalization error the set of input patterns, after sub-sampling, windowing and joining, were shuffled and split into a training and a test set; such a procedure was repeated 10 times to make up for statistical fluctuations. The eventual generalization performance was worked out by averaging the classification errors measured in the 10 runs. The overall system accuracy was evaluated by combining the accuracy measures of the pipelined classifiers, in compliance with the hierarchical schema described in the previous Section.

An additional goal of the tests was to assess the influence of the window size, L , on the accuracy of the classifiers. Therefore, the experiments covered a set of possible window sizes: $L \in \{10, 50, 100, 200\}$. Tables 4.32 and 4.33 give the number of patterns and the distribution of classes $\{-1, +1\}$ for the signal classifier and the detection-event classifier, respectively. The Tables report the cardinalities of the data sets used for training and test.

The following Sections report on the obtained results for the data-analysis process, and give the classification performances for each neural classifiers implemented.

4.6.4.2 Random Projection Based Data Visualization

In all graphs, cross marks and black circles mark patterns associated with opposite classes $(-1, +1)$. A promising projection is attained when the patterns belonging to one class appear to be easily separated from those belong-

Event Classifier Window size, L	Training set			Test set		
	-1	+1	Total	-1	+1	Total
10	123	142	265	106	145	251
50	123	142	265	106	144	250
100	123	142	265	106	144	250
200	123	142	265	106	144	250

Table 4.33: Number of patterns for the Event Classifier

ing to the other.

In the case of the signal classifier (Fig.4.28), the graph shows that only a limited portion of patterns tended to spread out when the windows size increased. By contrast, a marked confusion among the two classes emerged when projecting the event-classifier data (Fig. 4.29).

In any case, the 2-dimensional views resulting from the random-projection analysis contributed in clarifying the data distribution and the class peculiarities that could not be observed in the original, time-domain space.

4.6.4.3 Plastic Neural Gas Experiments

This set of experiments aimed at exploring the capabilities of an unsupervised paradigm to analyze the distribution of classes in the target system. The relative advantage of an unsupervised, Vector-Quantization model mostly lies in the limited complexity of the classifier model, which in turn sharply constrains the bounds to the expected worst-case classification error [3]. For coherence with the previous results, the PGAS model supported the Signal/Event hierarchical strategy.

The graphs in Fig 4.31 a)-c) give the errors that were measured when using the PGAS results to support VQ-based classification [76] for the Signal-, Event- and Overall-classifier configuration, respectively. Experimental evidence showed that, in all cases, accuracy seemed to degrade when the window-overlap size increased. This might appear a non-intuitive outcome, since one might expect that increasing time correlation among patterns would enhance classification accuracy on average.

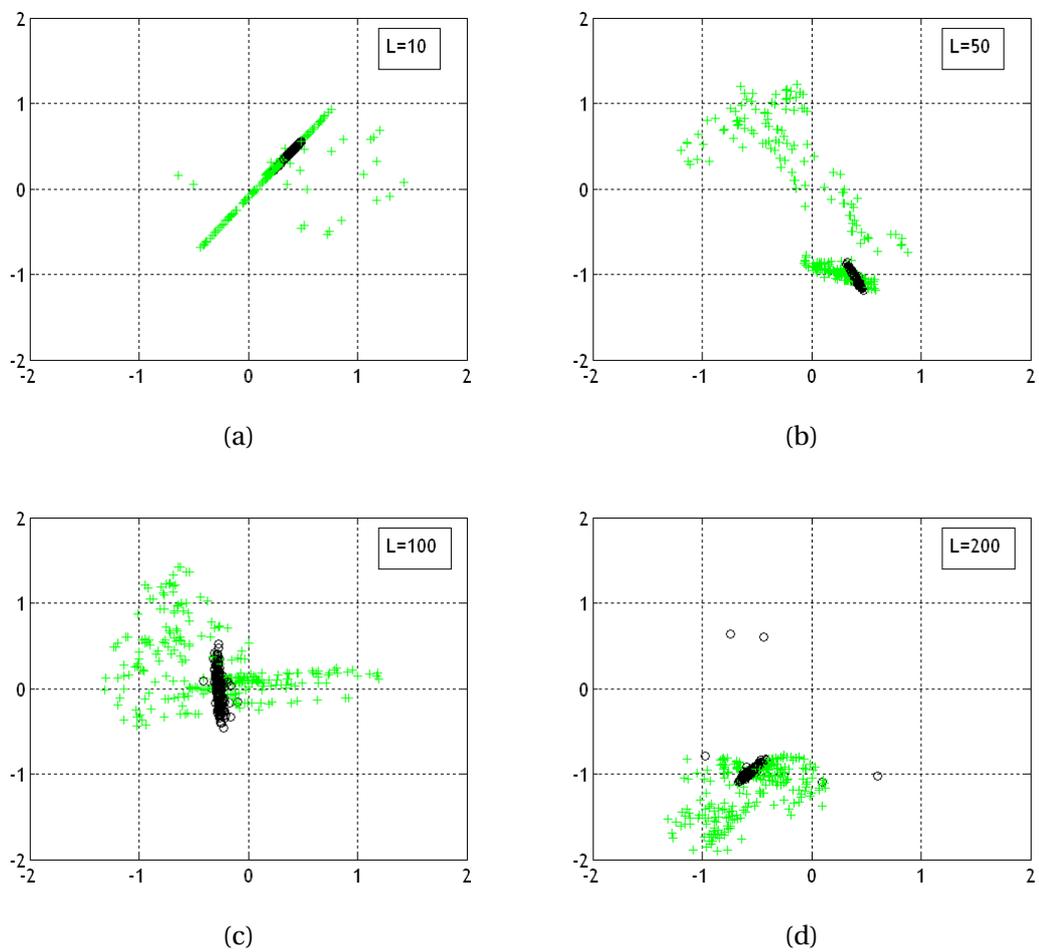


Figure 4.28: Signal-classifier data analysis: 2-dim Random Projections for increasing window overlap, L

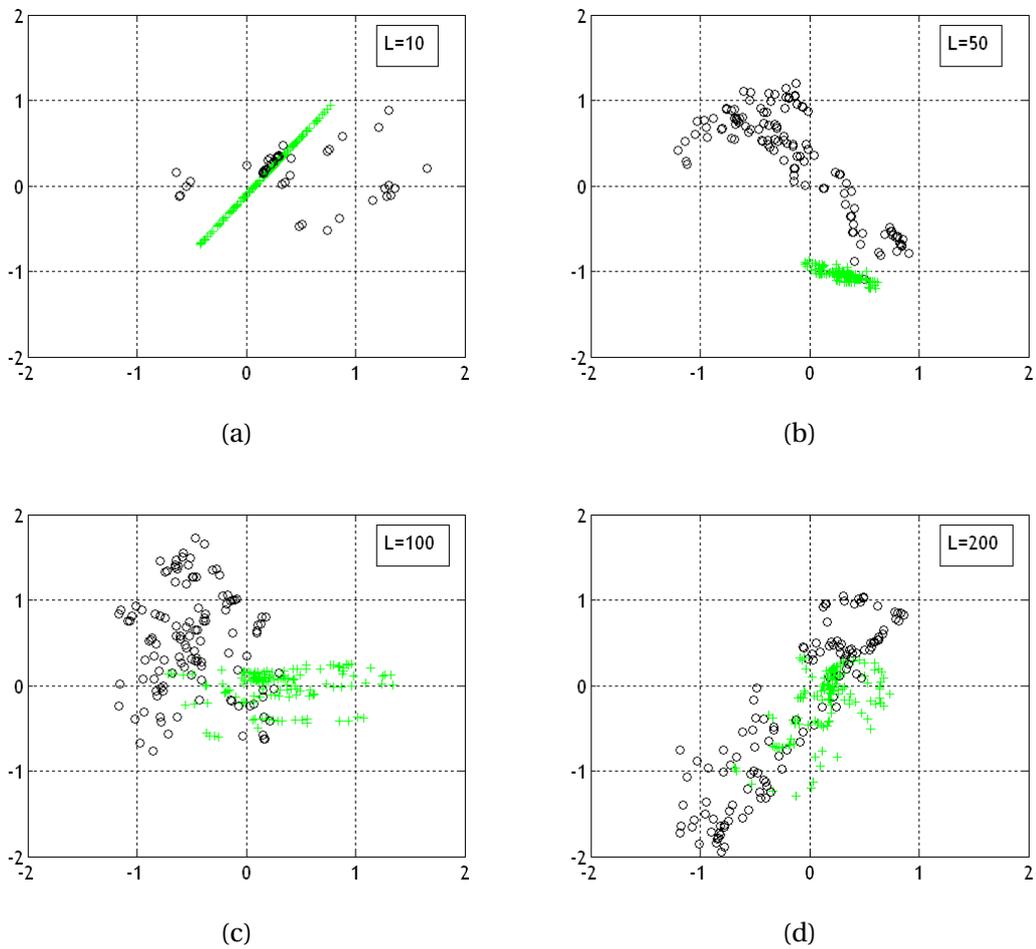


Figure 4.29: Event-classifier data analysis: 2-dim Random Projections for increasing window overlap, L

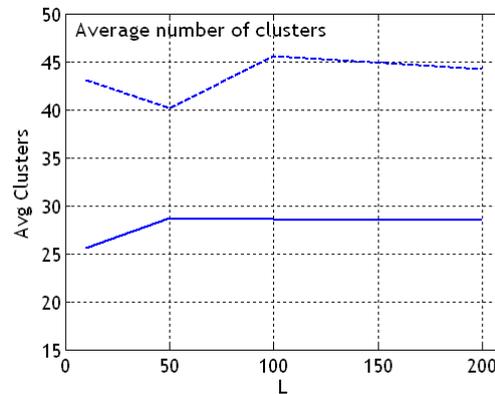


Figure 4.30: Unsupervised data analysis: optimal number of clusters vs increasing window overlap, L (dashed line indicates Signal-classifier data, the solid line indicates Event-classifier data)

One may get a direct explanation of such empirical finding by considering that the PGAS model operates in the original space of data, and does not imply any pattern mapping onto any projection or feature space, hence overlapped pattern might suffer from an increased amount of noise. The SVM and CBP models, instead, typically apply a non-linear mapping of data, thus notably improving the resulting generalization ability.

An additional explanation of the better performances of the latter models might stem from their intrinsic skill at coping with the curse of dimensionality. This is especially true for the SVM model, irrespectively of the specific kernel expression adopted.

4.6.4.4 Support Vector Machine Experiments

The implemented SVM training procedure was based on the SMO algorithm and used the classical settings, namely, a first-order selection strategy of the working set and a tolerance setting $\tau = 10^{-3}$ in the Karush-Kuhn-Tucker optimality conditions; the code was implemented as a Matlab C-coded mex-file routine.

Figures 4.34, 4.35 give the test errors (with confidence intervals) that were measured when training SVM classifiers with the optimal settings of kernel parameters (C, σ) obtained from the previous analysis, and varying the window-

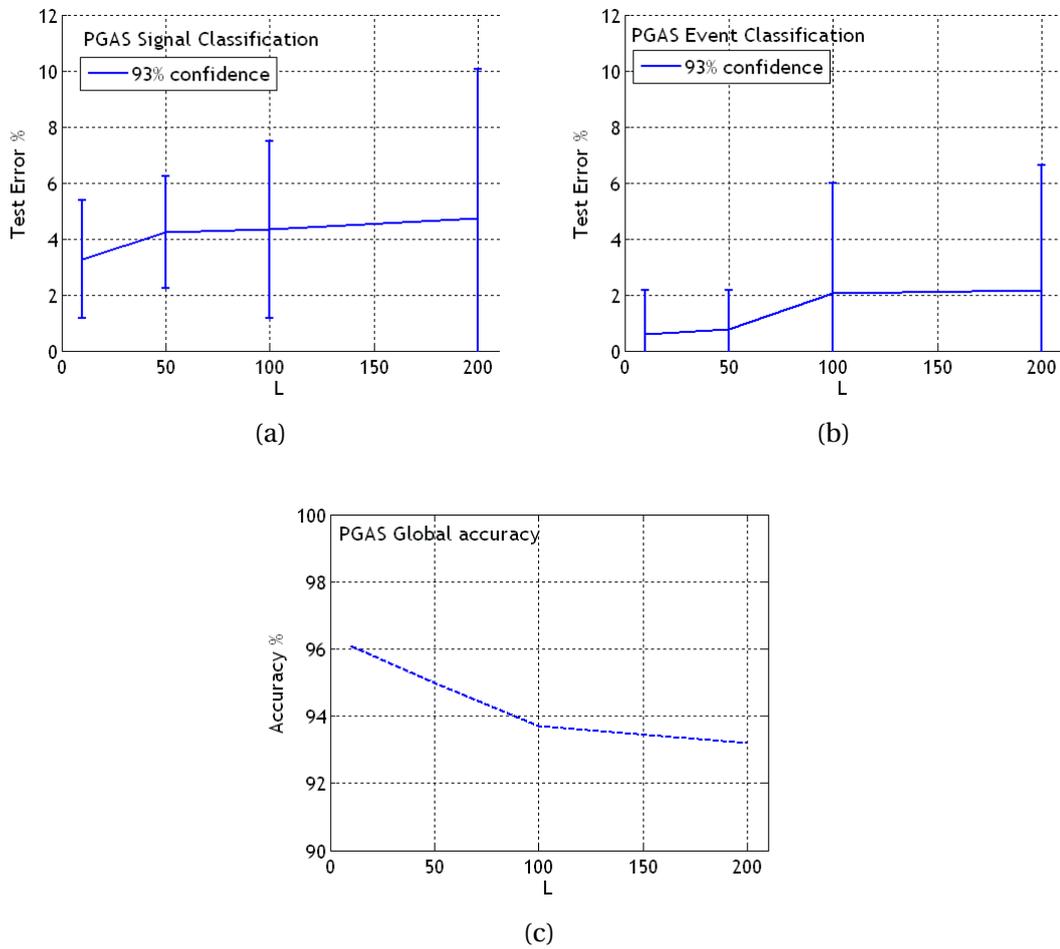


Figure 4.31: PGAS classification performances; a) - Signal classifier; b) - Event classifier; c) - Overall System performance

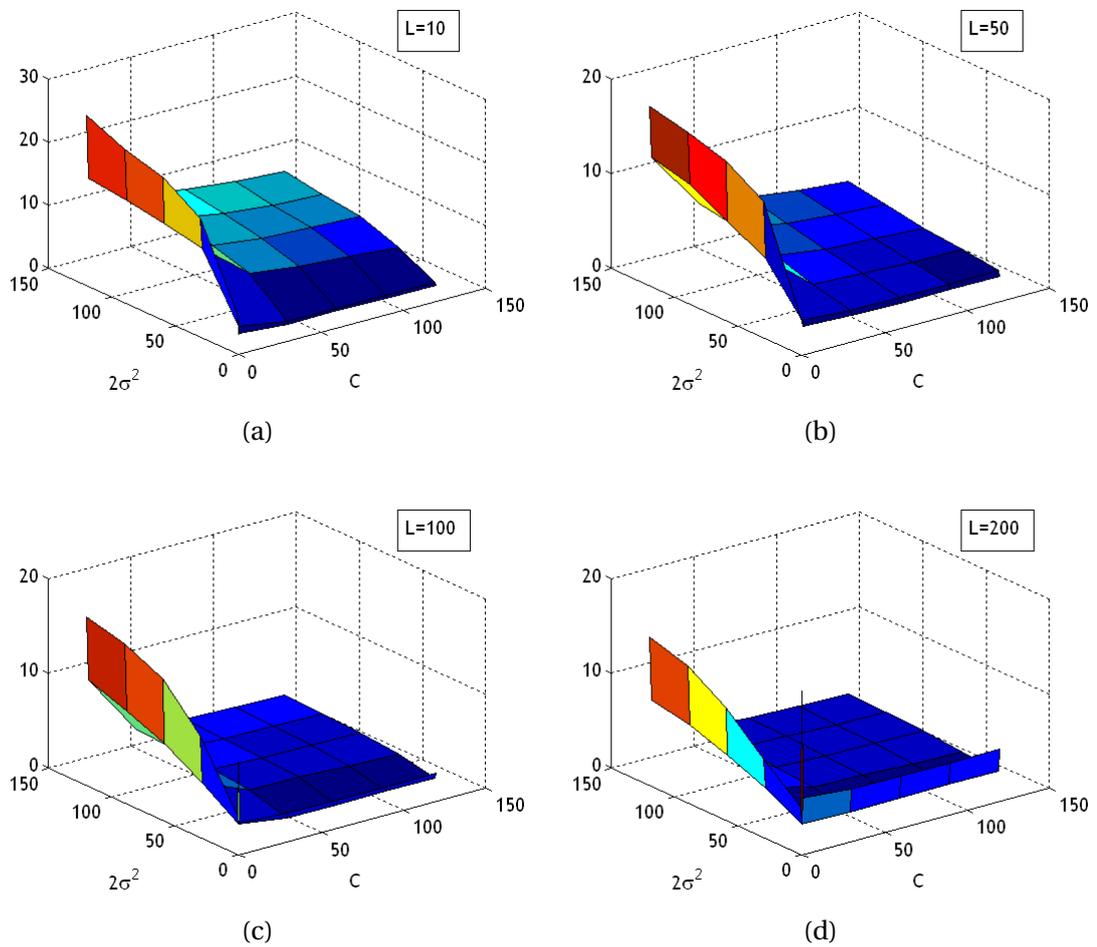


Figure 4.32: Signal classifier: measured test errors for different kernel parameters

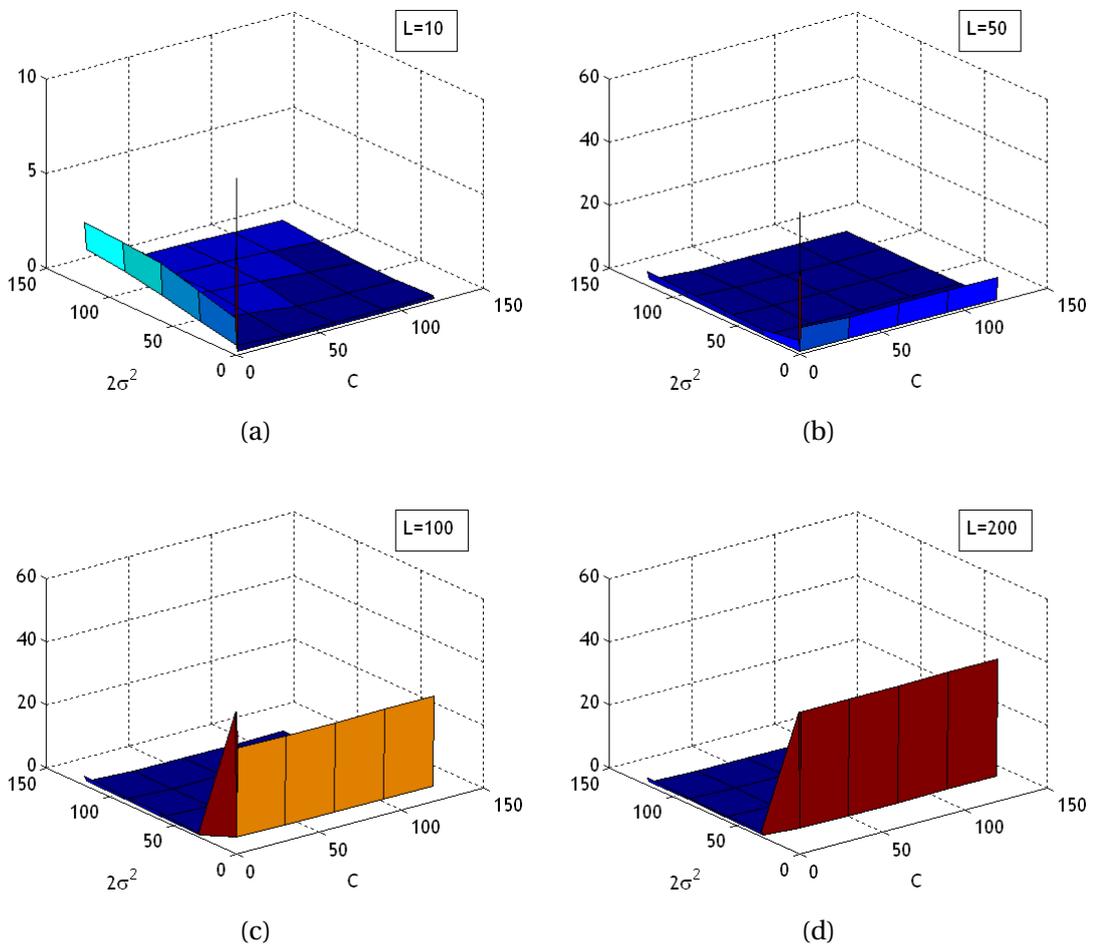


Figure 4.33: Event classifier: measured test errors for different kernel parameters

overlap size, L . Figure 4.36 shows the resulting overall accuracy. The analysis of the above results pointed out that, when using the SVM model, the size of the window was not a critical parameter for both the Signal and the Event Classifier. Empirical evidence also showed that the Event-Classification task seemed considerably simpler than the Signal-Classification one. This was especially true when considering the sensitivity to the kernel parameters; in the case of Event Classification, a wide range of parameters yielded good classification results, whereas such a robust behavior was not detected in the other case.

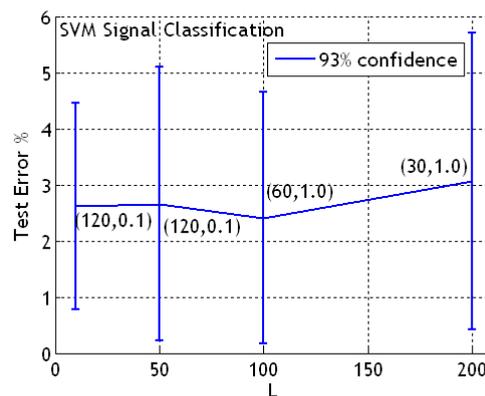


Figure 4.34: Signal Classifier: test errors for optimal kernel par.

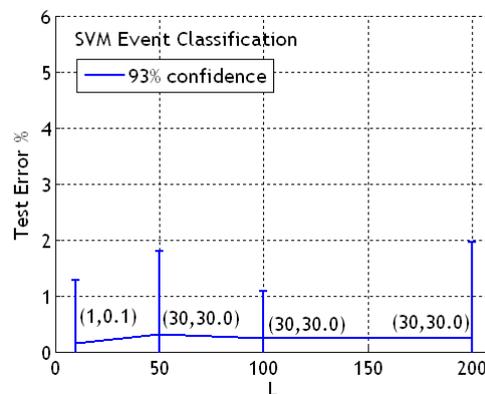


Figure 4.35: Event Classifier: test errors for optimal kernel par.

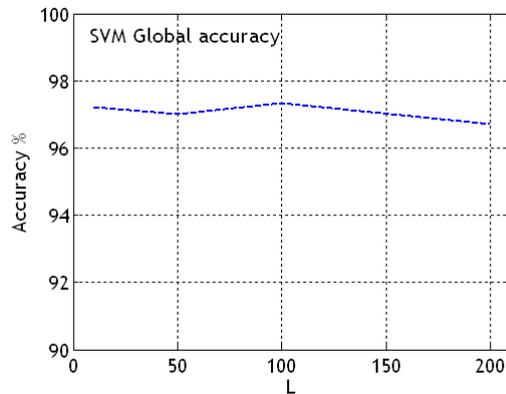


Figure 4.36: Global classification accuracy for optimal kernel par. and varying window size

4.6.4.5 Circular Back Propagation Experiments

The set of experiments involving the Circular Back Propagation networks aimed at assessing the validity of the general detection schema, by testing an alternative model in the Signal- and the Event-classifier modules. The experiments again gave a measure of the generalization ability of the tested classifiers. The analysis was carried out by testing two different topology configurations of the multilayer architecture: one configuration included 5 neurons in the hidden layer, whereas the second configuration included 10 neurons. To obtain a robust estimate of the run-time error, the training process was iterated three times and the resulting performances were averaged.

Figure 4.37 a)-d) give the obtained results in terms of classification accuracy, while Figure 4.38 shows the measured detection accuracy associated with the overall detection system. An interesting element of the obtained results is that CBP-based classifiers seemed to be less sensitive to the network parameters (i.e., the number of hidden neurons) than SVM-based classifiers, and overall yielded a comparable, sometimes even better, accuracy on test data.

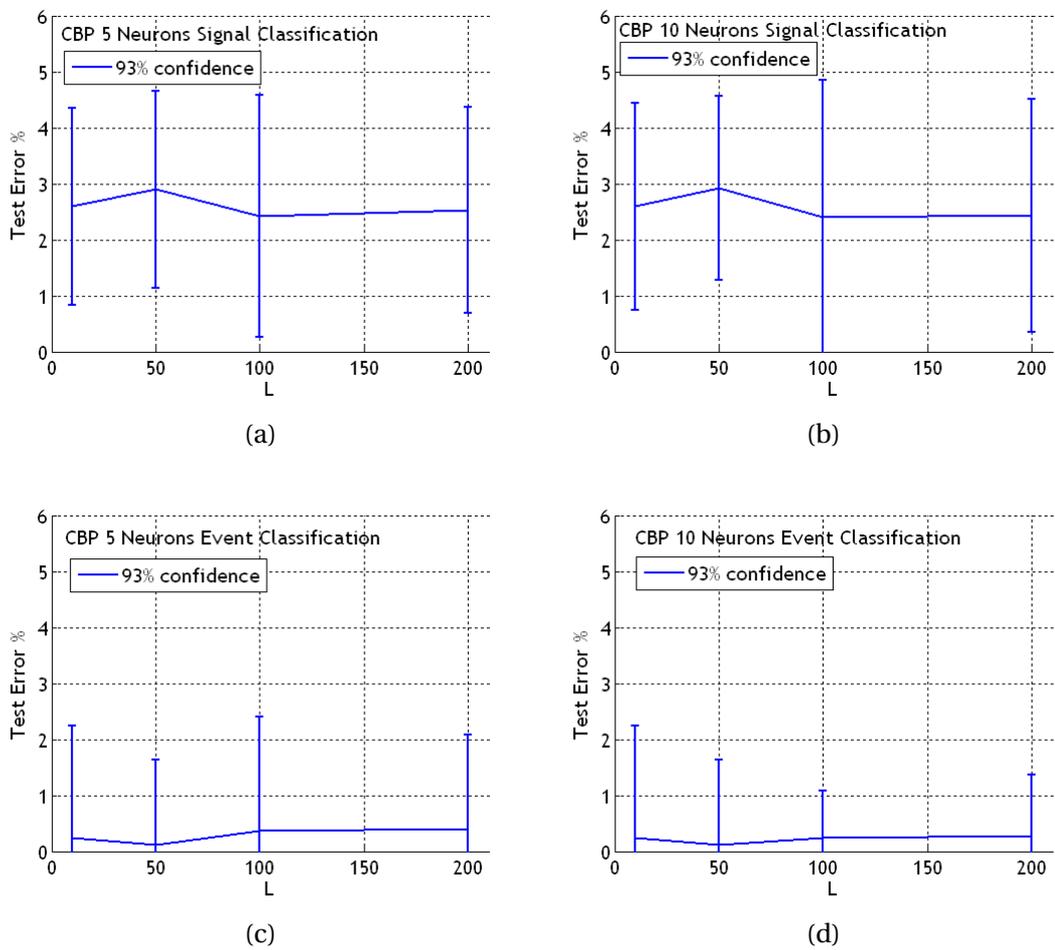


Figure 4.37: CBP, Signal/Event Accuracies

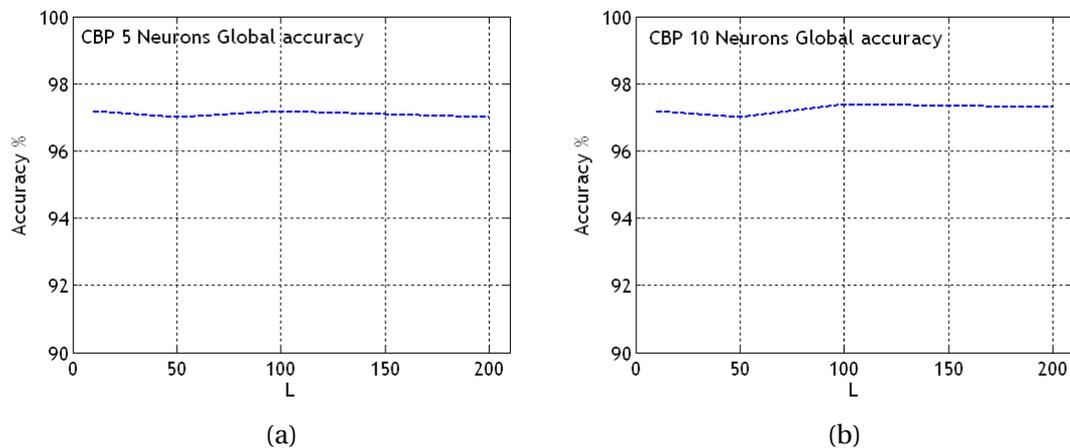


Figure 4.38: Overall detection performances when using CBP networks

4.6.4.6 Discussion on the eventual detection system

In principle, one might argue that the accuracy attained by the integrated system on the test tests did not reach the optimal level 100% and the system was subject to a few false alarms; in practice, however, this conclusion holds only when considering individual patterns. If instead one takes into account the sequential progression of empirical data, one verifies that misclassification events only occur at the transitions between normal operation and intrusion detections (or viceversa). In other words, it can be predicted theoretically, and was verified experimentally, that all misclassified patterns lied within the time intervals that spanned the beginning (or the end) of an intrusion signal. As soon as the window covered a sufficient portion of the signal associated with an intrusion (or with a normal situation), the classification always proved steadily successful and complete. Such a behavior was due to the fact that, during transitions, the overlapped-window patterns involved signal samples belonging to different classes (in both the Signal and the Event classifiers), and therefore carried a degree of inherent ambiguity. The practical consequence of this phenomenon is that the system yielded a 100% accuracy, provided a delay time was allowed between the start of a change in status and the associate prompting by the system. The delay never exceeded the size, L , of the window and on average was reasonably set to the time span

covering three patterns; in a 50-Hz sampling system, this amounted to 6 ms at most.

To perform a technical choice between the alternative classifier models for the detection system configuration, one should take into account the following aspects:

1. Random Projections showed that the complexity of each classification problem involved increases when the dimensionality of the space increases.
2. Both the SVM and the CBP model exhibited remarkable generalization abilities as opposed to the PGAS model, which did not attain a comparable accuracy. It was reasonable to expect this when considering that PGAS classifiers stemmed from an unsupervised training process.

As a consequence of those issues, one could conclude that the SVM and/or the CBP model seemed the most promising choices for further implementations on embedded platforms. The former approach, in particular, appeared in fact preferable in view of its lesser dependency on the cardinality of the input space.

These considerations motivated an additional analysis of the SVM performances to further explore the false positive rates yielded by the trained classifiers. The graph in figure 4.39 shows that the Signal-Classification problem proved more complex than the Event-Classification one: the false-positive rate always kept smaller than 2%, thus yielding a low rate of false alarms. In addition, the false-positive rate also remained low in the case of small window-overlap sizes, since a false-positive rate of 1% was measured for an window size $L = 10$ sample. This meant that such a window size was a reasonable choice for the final system and well matched the evidence obtained from the analysis with Random Projections. Similar conclusions were reached when observing the false-positive rates yielded by CBP-based classifiers.

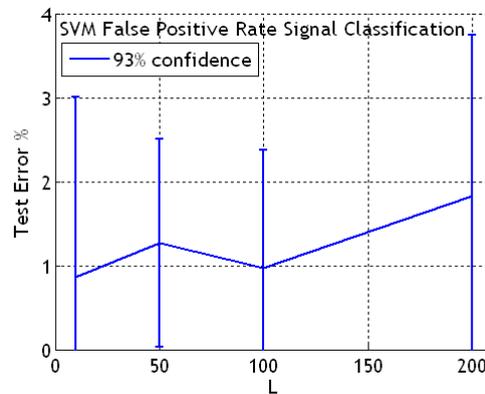


Figure 4.39: Measured false-positive rates for SVM classifiers

4.6.5 Conclusions

The paper presented an automatic system for intrusion detection and port protection, based on machine-learning methods. The obtained results confirmed that the proposed approach represented a viable solution to the real-time detection problem in the presence of real scenarios under non-ideal operative conditions.

In particular, the paper showed that the adaptive methodology made it possible to overcome the crucial issues of possibly defective sensing components and imperfect synchronization of signals from multiple sensors. The hierarchical approach integrating SVM/CBP-based classifiers yielded a global detection accuracy higher than 97% on real data in noisy environments, with a false positive rate smaller than 1%.

A remarkable point of interest of the overall approach is that the specific models of learning machines adopted in the presented research only relies on non-linear basis functions and Hilbert dot products. The associate analytical framework proves that the involved computations can be efficiently performed on inexpensive embedded architectures. This opens new vistas on the development of a fully embedded intrusion detection system. Future research lines will also investigate alternative learning approaches (i.e. alternative learning models and pre-processing methods) for further enhancing detection performances.

5

Conclusions

This thesis has address both theoretical problems and engineering applications of statistical learning theory.

From the theoretical point of view this thesis mainly contributed with two analysis concerning the generalization of the usual regularization term $\|f\|_{\mathcal{H}}^2$ in kernel methods. The first analysis, carried after a preliminary study on the regularized mean problem, consists in imposing a, possibly, *oracular* regularization operator such as $\|Tw\|^2$, the second is using biased regularization $\|f - f_0\|_{\mathcal{H}}^2$. Both these strategies shape the regularization term in a specific way. The first solution, studied for Tikhonov regularization, leads to the formal possibility of unifying concepts from learning, regularization, filtering and shrinking disciplines; from the practical point of view the *oracular* form of T is a starting point from which one can work out practical approximations of the best possible regularization operator.

The second approach, based on biased regularization, allows an easy embedding of prior information in kernel methods and neural networks. In particular when, as in this case, the reference model f_0 is consistent with the structure of the input data one obtains both a semi supervised learning method and tight generalization bounds; this last feature is due to the sharp reduction of the hypothesis space induced by f_0 when a clustering hypothesis holds true. The immediate practical utility is that one can perform reliable model selection in semi supervised learning by using generalization bounds and a very low number of labeled data. The applicative domain of

this method is in Digit Image Classification, Text Classification and in general in every domain where alternatively a clustering or manifold hypothesis holds.

The proposed method of centering the hypothesis space around f_0 is general and reference functions different from the one provided by clustering can be used. The reference function f_0 can be considered as an a priori suggestion when performing the learning step; in general it has been shown that also a non optimal f_0 leads to improvements in final accuracy and in the amount of shrinking of the hypothesis space. The key idea under f_0 is that one should always force a sort of *ordering* on the space of functions; the hypothesis $f_0 = 0$, usually used in kernel methods, is very far from any real learned function, for this reason the *distance* from f_0 and f_{true} is usually high; instead if one has a guess f_0 then a shorter *path* is needed to reach f_{true} . Indeed one should consider that the complexity of a class of functions is not an absolute concept, but a relative one; f_0 determines how much the learning process has to change parameters in order to reach f_{true} . Vapnik theory clearly states that complexity is due to the *distance* from the a priori to the final learned function; a suitably chosen f_0 can shorten this distance considerably.

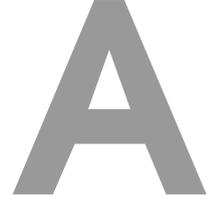
Concerning the last contribution of Chapter 3 a brief note discussed the differences between semi supervised learning and transduction: in particular a simple proof of an explicit transductive generalization bound was given were it is explicitly shown that transduction is simpler than induction.

Lastly several engineering applications and algorithmic improvements were discussed and developed in Chapter 4.

There are several open problems: the first regards the possibility of finding f_0 ; it remains open the problem of finding a good f_0 in highly non linear small sample problems; in large sample problems finding f_0 it is easy because one can split the data in two folds; from the first one learns f_0 from the second one learns the final function f . This strategy in small sample problems is not practical and not reliable; for this reason finding f_0 in such learning tasks is still an open problem. A further extension of the given results regards developing an oracular regularization machinery for kernel methods; formally this consists in defining oracular regularizers in kernel space instead of the

original space as it has been performed for Tikhonov regularization.

Appendices



Regularized Mean Problem

A.1 Proof of 3.1.1

$$\sigma_{\bar{x}}^2 \equiv \mathbf{E}_X \{(\bar{x} - \mathbf{E}_X \{\bar{x}\})^2\} = \mathbf{E}_X \{\bar{x}^2\} - \mathbf{E}_X \{\bar{x}\}^2$$

The first quantity of the right hand side can be deduced by:

$$\begin{aligned} \mathbf{E}_X \{\bar{x}^2\} &= \lambda^2 \mathbf{E}_X \{\bar{x}_0^2\} = \frac{\lambda^2}{m^2} \mathbf{E}_X \left\{ \sum_{i=1}^m x_i \sum_{j=1}^m x_j \right\} \\ &= \frac{\lambda^2}{m^2} \left[\mathbf{E}_X \left\{ \sum_{i=1}^m x_i^2 \right\} + \mathbf{E}_X \left\{ \sum_{i=1}^m \sum_{j \neq i}^m x_j x_i \right\} \right] = \\ &= \frac{\lambda^2}{m^2} \left[\sum_{i=1}^m \mathbf{E}_X \{\mu^2 + \sigma^2\} + m(m-1)\mu^2 \right] = \\ &= \frac{\lambda^2(\sigma^2 + m\mu^2)}{m} \end{aligned}$$

Then it follows:

$$\sigma_{\bar{x}}^2 = \frac{\lambda^2(\sigma^2 + m\mu^2)}{m} - (\lambda\mu)^2 = \frac{\lambda^2\sigma^2}{m}$$

Q.D.E.

A.2 Proof of the variance, 3.1.2

Let consider the quantity: $\mathbf{E}_X \left\{ \sum_{i=1}^m (x_i - \bar{x})^2 \right\}$. This becomes:

$$\mathbf{E}_X \left\{ \sum_{i=1}^m (x_i - \lambda\bar{x}_0)^2 \right\} = \sigma^2 \left[(m\mu^2/\sigma^2 + 1)(\lambda - 1)^2 + (m - 1) \right]$$

Then $\sigma^2 = \mathbf{E}_X \left\{ \frac{\sum_{i=1}^m (x_i - \bar{x})^2}{(m\mu^2/\sigma^2 + 1)(\lambda - 1)^2 + (m - 1)} \right\}$. This last expression means that $\frac{\sum_{i=1}^m (x_i - \bar{x})^2}{(m\mu^2/\sigma^2 + 1)(\lambda - 1)^2 + (m - 1)}$ is an unbiased estimator of σ^2 , consequently the number of degrees of freedom is $(m\mu^2/\sigma^2 + 1)(\lambda - 1)^2 + (m - 1)$.

Q.D.E

A.3 Proof of degrees of freedom, 3.1.1

Recalling the general expression of $\mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\} - \sigma^2/m$ one has:

$$\begin{aligned} & \mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\} - \sigma^2/m = \\ & = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \int_{-\gamma}^{+\gamma} \underbrace{[\lambda(X)\bar{x}_0 - \mu]^2}_B \underbrace{\left[\frac{1}{2\gamma} \frac{1}{(\sigma\sqrt{2\pi})^m} \prod_{i=1}^m \exp \left[\frac{-(x_i - \mu)^2}{2\sigma^2} \right] \right]}_D d\mu dx_1 \dots dx_m (&) \end{aligned}$$

As a first step the term B can be written as: $[\lambda(X)\bar{x}_0 - \bar{x}_0 + \bar{x}_0 - \mu]^2 = (\lambda(X) - 1)^2 \bar{x}_0^2 + 2(\lambda(X) - 1)\bar{x}_0(\bar{x}_0 - \mu) + (\bar{x}_0 - \mu)^2 \triangleq B_1 + B_2 + B_3$ Further, denoting by S^2 the unbiased estimator of the variance, it is easy to note that:

$\sum_{i=1}^n (x_i - \mu)^2 = (m - 1)S^2 + m(\bar{x}_0 - \mu)^2$ The part of the integral (&) pertinent to B_3 is:

$$\begin{aligned} & \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \int_{-\gamma}^{+\gamma} B_3 D d\mu dx_1 \dots dx_m = \\ & = \int_{-\gamma}^{+\gamma} \frac{1}{2\gamma} d\mu \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} (\bar{x}_0 - \mu)^2 \frac{1}{(\sigma\sqrt{2\pi})^m} \prod_{i=1}^m \exp \left[\frac{-(x_i - \mu)^2}{2\sigma^2} \right] dx_1 \dots dx_m = \frac{\sigma^2}{m} \end{aligned}$$

Recalling the algebraic independence of λ from μ , the remaining parts are:

$$\begin{aligned} & \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \int_{-\gamma}^{+\gamma} (B_1 + B_2) D d\mu dx_1 \dots dx_m = \\ & \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \frac{\sigma\sqrt{(2\pi)/m}}{(\sigma\sqrt{2\pi})^m} \exp \left[\frac{-(m-1)S^2}{2\sigma^2} \right] \left\{ (\lambda - 1)^2 \bar{x}_0^2 \frac{1}{2\gamma} \int_{-\gamma}^{+\gamma} \frac{1}{\sigma\sqrt{(2\pi)/m}} \exp \left[\frac{-(\bar{x}_0 - \mu)^2}{2\sigma^2/m} \right] d\mu + \right. \\ & \left. + 2(\lambda - 1)\bar{x}_0 \frac{1}{2\gamma} \int_{-\gamma}^{+\gamma} (\bar{x}_0 - \mu) \frac{1}{\sigma\sqrt{(2\pi)/m}} \exp \left[\frac{-(\bar{x}_0 - \mu)^2}{2\sigma^2/m} \right] d\mu \right\} d\mu dx_1 \dots dx_m \end{aligned}$$

Then by integrating on μ :

$$\int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \frac{\sigma\sqrt{(2\pi)/m}}{(\sigma\sqrt{2\pi})^m} \exp\left[\frac{-(m-1)S^2}{2\sigma^2}\right] \left\{ (\lambda - 1)^2 \bar{x}_0^2 \frac{1}{4\gamma} \left[\operatorname{erf}\left(\frac{\bar{x}_0+\gamma}{\sigma\sqrt{2/m}}\right) - \operatorname{erf}\left(\frac{\bar{x}_0-\gamma}{\sigma\sqrt{2/m}}\right) \right] + \right. \\ \left. + (\lambda - 1) \bar{x}_0 \frac{1}{\gamma} \frac{\sigma}{\sqrt{2\pi m}} \left[\exp\left(-\left[\frac{\bar{x}_0-\gamma}{\sigma\sqrt{2/m}}\right]^2\right) - \exp\left(-\left[\frac{\bar{x}_0+\gamma}{\sigma\sqrt{2/m}}\right]^2\right) \right] \right\} dx_1 \dots dx_m \quad (\&2)$$

Taking the limit $\gamma \rightarrow \infty$ of (&2) one has to observe three facts:

1. For the first addend one has that $\operatorname{erf}\left(\frac{\bar{x}_0+\gamma}{\sigma\sqrt{2/m}}\right) - \operatorname{erf}\left(\frac{\bar{x}_0-\gamma}{\sigma\sqrt{2/m}}\right) = \operatorname{erf}\left(\frac{\bar{x}_0+\gamma}{\sigma\sqrt{2/m}}\right) + \operatorname{erf}\left(\frac{\gamma-\bar{x}_0}{\sigma\sqrt{2/m}}\right)$ and taking the limit $\gamma \rightarrow \infty$, this last expression tends to 1.

Globally $(\lambda - 1)^2 \bar{x}_0^2 \frac{1}{4\gamma} \left[\operatorname{erf}\left(\frac{\bar{x}_0+\gamma}{\sigma\sqrt{2/m}}\right) - \operatorname{erf}\left(\frac{\bar{x}_0-\gamma}{\sigma\sqrt{2/m}}\right) \right]$ tends to 0 by positive values.

2. The second term $(\lambda-1) \bar{x}_0 \frac{1}{\gamma} \frac{\sigma}{\sqrt{2\pi m}} \left[\exp\left(-\left[\frac{\bar{x}_0-\gamma}{\sigma\sqrt{2/m}}\right]^2\right) - \exp\left(-\left[\frac{\bar{x}_0+\gamma}{\sigma\sqrt{2/m}}\right]^2\right) \right]$ tends to 0 faster then the first addend already analyzed. Note that $\exp\left(-\left[\frac{\bar{x}_0-\gamma}{\sigma\sqrt{2/m}}\right]^2\right) - \exp\left(-\left[\frac{\bar{x}_0+\gamma}{\sigma\sqrt{2/m}}\right]^2\right)$ tends to 0 by itself, also without the term $1/\gamma$.

3. The asymptotic behavior of the two addends is dominated by the first one. Therefore one gets:

$$\lim_{\gamma \rightarrow \infty} \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \frac{\sigma\sqrt{(2\pi)/m}}{(\sigma\sqrt{2\pi})^m} \exp\left[\frac{-(m-1)S^2}{2\sigma^2}\right] (\lambda - 1)^2 \bar{x}_0^2 \frac{1}{4\gamma} \left[\operatorname{erf}\left(\frac{\bar{x}_0+\gamma}{\sigma\sqrt{2/m}}\right) - \operatorname{erf}\left(\frac{\bar{x}_0-\gamma}{\sigma\sqrt{2/m}}\right) \right] dx_1 \dots dx_m = 0$$

Where the integral goes to 0 from positive values.

Then :

$$\int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \int_{-\gamma}^{+\gamma} (B_1 + B_2 + B_3) D d\mu dx_1 \dots dx_m = 0 + \frac{\sigma^2}{m}$$

by which immediately follows point a) of the theorem.

Taking the limit $\gamma \rightarrow 0$ of (&2):

$$\begin{aligned} & \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \frac{1}{(\sigma\sqrt{2\pi})^m} \exp \left[\frac{-(m-1)S^2}{2\sigma^2} \right] \bar{x}_0^2 \\ & \exp \left(- \left[\frac{\bar{x}_0}{\sigma\sqrt{2/m}} \right]^2 \right) \{(\lambda - 1)^2 + 2(\lambda - 1)\} dx_1 \dots dx_m = \\ & = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \frac{1}{(\sigma\sqrt{2\pi})^m} \exp \left[\frac{-(m-1)S^2}{2\sigma^2} \right] \bar{x}_0^2 \\ & \exp \left(- \left[\frac{\bar{x}_0}{\sigma\sqrt{2/m}} \right]^2 \right) \{(\lambda - 1)(\lambda + 1)\} dx_1 \dots dx_m \end{aligned}$$

By hypothesis $\lambda < 1$ then the above integral is less than 0.

Hence

$$\int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \int_{-\gamma}^{+\gamma} (B_1 + B_2 + B_3)D d\mu dx_1 \dots dx_m < \frac{\sigma^2}{m}$$

by which follows point b).

Point c) follows from the continuous dependence of the integral (&) on γ parameter and from points a) and b).

In particular point a) shows that for $\gamma \rightarrow \infty$ then $\mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\} - \sigma^2/m \rightarrow 0$ from positive values; point b) shows that for $\gamma \rightarrow 0$ then $\mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\} - \sigma^2/m < 0$. Given the continuity of $\mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\} - \sigma^2/m$ on γ then it must exist at least a value $\gamma = \hat{\gamma}$ where $\mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\} - \sigma^2/m = 0$. Moreover for $\gamma < \hat{\gamma}$, the function $\mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\} - \sigma^2/m$ is always negative and, conversely, for $\gamma > \hat{\gamma}$ $\mathbf{E}_\mu^\gamma \mathbf{E}_X \{(\lambda(X)\bar{x}_0 - \mu)^2\} - \sigma^2/m$ is always positive.

Q.D.E

A.4 Proof of 3.1.3

The general expression is:

$$\int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \int_{-\gamma}^{+\gamma} [\lambda(X)\bar{x}_0 - \mu]^2 \left[\frac{1}{2\gamma} \frac{1}{(\sigma\sqrt{2\pi})^m} \prod_{i=1}^m \exp \left[\frac{-(x_i - \mu)^2}{2\sigma^2} \right] \right] d\mu dx_1 \dots dx_m (&)$$

Recalling that, in this case, $\lambda(X) = \lambda^{orac}$, and thus it is not sample dependent, then previous expression can be written as:

$$\begin{aligned} & \int_{-\gamma}^{+\gamma} \frac{1}{2\gamma} [\mathbf{E}_X \{ \bar{x}_0^2 (\lambda^{orac} - 1)^2 \} + \mathbf{E}_X \{ 2\bar{x}_0 (\lambda^{orac} - 1) (\bar{x}_0 - \mu) \} + \mathbf{E}_X \{ (\bar{x}_0 - \mu)^2 \}] d\mu = \\ & = \int_{-\gamma}^{+\gamma} \frac{1}{2\gamma} [(\lambda^{orac} - 1)^2 (\mu^2 + \sigma^2/m) + 2(\lambda^{orac} - 1)(\sigma^2/m) + (\sigma^2/m)] d\mu \end{aligned}$$

Plugging the oracular regularizer one gets:

$$\begin{aligned} & \int_{-\gamma}^{+\gamma} \frac{1}{2\gamma} \left[\left(\frac{\sigma^2/m}{\mu^2 + \sigma^2/m} \right)^2 (\mu^2 + \sigma^2/m) - 2 \left(\frac{\sigma^2/m}{\mu^2 + \sigma^2/m} \right) (\sigma^2/m) + (\sigma^2/m) \right] d\mu = \\ & = - \int_{-\gamma}^{+\gamma} \frac{1}{2\gamma} \frac{(\sigma^2/m)^2}{\mu^2 + \sigma^2/m} d\mu + (\sigma^2/m) = \end{aligned}$$

Then $= - \int_{-\gamma}^{+\gamma} \frac{1}{2\gamma} \frac{(\sigma^2/m)^2}{\mu^2 + \sigma^2/m} d\mu + (\sigma^2/m) < (\sigma^2/m)$ that proves the thesis

Q.D.E

A.5 Proof of 3.1.4

The general two-sided expression is as per (3.18): the one sided version clearly is:

$$\int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \int_0^{+\gamma} [\lambda(X)\bar{x}_0 - \mu]^2 \left[\frac{1}{\gamma} \frac{1}{(\sigma\sqrt{2\pi})^m} \prod_{i=1}^m \exp \left[\frac{-(x_i - \mu)^2}{2\sigma^2} \right] \right] d\mu dx_1 \dots dx_m$$

after steps analogous to Theorem 1 proof one gets the above integral equals:

$$\begin{aligned} & \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \frac{\sigma\sqrt{(2\pi)/m}}{(\sigma\sqrt{2\pi})^m} \exp \left[\frac{-(m-1)S^2}{2\sigma^2} \right] \left\{ (\lambda - 1)^2 \bar{x}_0^2 \frac{1}{\gamma} \int_0^{+\gamma} \frac{1}{\sigma\sqrt{(2\pi)/m}} \exp \left[\frac{-(\bar{x}_0 - \mu)^2}{2\sigma^2/m} \right] d\mu + \right. \\ & \left. + 2(\lambda - 1)\bar{x}_0 \frac{1}{\gamma} \int_0^{+\gamma} (\bar{x}_0 - \mu) \frac{1}{\sigma\sqrt{(2\pi)/m}} \exp \left[\frac{-(\bar{x}_0 - \mu)^2}{2\sigma^2/m} \right] d\mu dx_1 \dots dx_m + \sigma^2/m \right\} \end{aligned}$$

If $\bar{x}_0 > 0$ then $\lambda = 1(\bar{x}_0) = 1$. In this case the last expression is equal to σ^2/m (this is the trivial non regularized case).

Conversely if $\bar{x}_0 < 0 \lambda = 1(\bar{x}_0) = 0$ then:

$$\int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \frac{\sigma\sqrt{(2\pi)/m}}{(\sigma\sqrt{2\pi})^m} \exp\left[\frac{-(m-1)S^2}{2\sigma^2}\right] \left\{ -\bar{x}_0^2 \frac{1}{\gamma} \int_0^{+\gamma} \frac{1}{\sigma\sqrt{(2\pi)/m}} \exp\left[\frac{-(\bar{x}_0-\mu)^2}{2\sigma^2/m}\right] d\mu + \right. \\ \left. -2|\bar{x}_0|^{\frac{1}{\gamma}} \int_0^{+\gamma} \mu \frac{1}{\sigma\sqrt{(2\pi)/m}} \exp\left[\frac{-(\bar{x}_0-\mu)^2}{2\sigma^2/m}\right] d\mu \right\} dx_1 \dots dx_m + \sigma^2/m$$

then the above integral is less than σ^2/m by which the thesis follows.

Q.D.E

A.6 Proof of Leave One Out 3.1.2

In order to obtain the minimum of (3.13) with respect to α one gets:

$$0 = \frac{\partial L_{loo}}{\partial \alpha} = \sum_{i=1}^m (x_i - \bar{\xi}_i) \frac{\sum_{i \neq k, k=1}^m x_k}{(m + \alpha - 1)^2} (*)$$

To further proceed the following identities must be taken into account:

$$x_i + \frac{x_i}{m+\alpha-1} = \frac{m+\alpha}{m+\alpha-1} x_i \\ \bar{\xi}_i + \frac{x_i}{m+\alpha-1} = \frac{\sum_{j=1}^m x_j}{m+\alpha-1}$$

Then:

$$x_i - \bar{\xi}_i = \frac{m + \alpha}{m + \alpha - 1} x_i - \frac{\sum_{j=1}^m x_j}{m + \alpha - 1}$$

Substituting in (*):

$$0 = \sum_{i=1}^m \left[(m + \alpha)x_i - \sum_{j=1}^m x_j \right] \left(\sum_{j=1}^m x_j - x_i \right)$$

That leads to:

$$\alpha^{loo} = \frac{m \sum_{i=1}^m x_i^2 - (\sum_{i=1}^m x_i)^2}{(\sum_{i=1}^m x_i)^2 - \sum_{i=1}^m x_i^2}$$

The corresponding λ^{loo} written in terms of the unbiased variance estimator

$$S^2 = \frac{1}{m-1} \sum_{i=1}^m (x_i - \bar{x}_0)^2 \text{ is :}$$

$$\lambda^{loo} = \frac{\bar{x}_0^2 - \frac{S^2}{m}}{\bar{x}_0^2}$$

Q.D.E

A.7 Proof of 3.1.5

Recalling notation:

$$\aleph(\xi, \tau, \varphi) \equiv \frac{\tau}{2} \sum_{i=1}^m (x_i - \xi)^2 + \frac{\varphi}{2} \xi^2 = \tau E_D + \varphi E_W$$

one has that: $E_W = \frac{1}{2} \xi^2$ and $E_D = \frac{1}{2} \sum_{i=1}^m (x_i - \xi)^2$.

Then, one gets:

$$\begin{aligned} E_D &= \frac{1}{2} \sum_{i=1}^m (x_i - \xi)^2 = \frac{1}{2} [\sum_{i=1}^m x_i^2 - 2\xi \sum_{i=1}^m x_i + m\xi^2] = \\ &= \frac{1}{2} [m(s^2 + \bar{x}_0^2) - 2m\bar{x}_0\xi + m\xi^2] = \\ &= \frac{m}{2} [s^2 + (\bar{x}_0 - \xi)^2] \end{aligned}$$

Where $s^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \bar{x}_0)^2$ is the biased estimator of the variance. From this follows:

$$\aleph(\xi, \tau, \varphi) = \tau \frac{m}{2} [s^2 + (\bar{x}_0 - \xi)^2] + \frac{\varphi}{2} \xi^2$$

Nullifying the derivative with respect to ξ is analogous to ask for the *maximum a posteriori* \bar{x}_{MP} :

$$\frac{\partial \aleph(\xi, \tau, \varphi)}{\partial \xi} = -m\tau(\bar{x}_0 - \bar{x}_{MP}) + \varphi \bar{x}_{MP} = 0$$

that leads to:

$$\bar{x}_{MP} = \frac{m\bar{x}_0\tau}{\varphi + m\tau}$$

Q.D.E

A.8 Proof of Maximal Evidence 3.1.3

Recalling notation:

$$\aleph(\xi, \tau, \varphi) \equiv \frac{\tau}{2} \sum_{i=1}^m (x_i - \xi)^2 + \frac{\varphi}{2} \xi^2 = \tau E_D + \varphi E_W$$

Assuming a Gaussian prior the following probabilities can be defined:

$$\begin{cases} p(\xi) \equiv \left(\frac{\varphi}{2\pi}\right)^{-1/2} \exp(-\varphi E_W) \\ p(D|\xi) \equiv \left(\frac{\tau}{2\pi}\right)^{-m/2} \exp(-\tau E_D) \\ p(D) \equiv \left(\frac{\varphi}{2\pi}\right)^{-1/2} \left(\frac{\tau}{2\pi}\right)^{-m/2} \int \exp(-\aleph) d\xi \end{cases}$$

where $p(D)$ is the evidence term. Using Bayes theorem:

$$p(\xi|D) = \frac{p(\xi)p(D|\xi)}{p(D)} = \frac{\exp(-\aleph)}{\int \exp(-\aleph) d\xi}$$

Indicating $\frac{\partial^2 \aleph}{\partial \xi^2} = \varphi + m\tau \equiv \vartheta$ one has: $\aleph = \aleph_{MP} + \frac{1}{2}(\xi - \bar{x}_{MP})^2 \vartheta$. Moreover $\int \exp(-\aleph) d\xi = \exp(-\aleph_{MP}) \left(\frac{2\pi}{\vartheta}\right)^{1/2}$.

Computing the logarithm of the evidence $p(D)$ leads to:

$$\ln(p(D)) = -\varphi E_W^{MP} - \tau E_D^{MP} - \frac{1}{2} \ln(\vartheta) + \frac{1}{2} \ln(\varphi) - \frac{m}{2} \ln(2\pi) + \frac{m}{2} \ln(\tau) (*)$$

This expression is a function on φ and τ . Then searching for the max of (*) one gets:

$$\begin{cases} \frac{\partial \ln(p(D))}{\partial \varphi} = 0 = -E_W^{MP} - \frac{1}{2\vartheta} + \frac{1}{2\varphi} \\ \frac{\partial \ln(p(D))}{\partial \tau} = 0 = -E_D^{MP} - \frac{m}{2\vartheta} + \frac{m}{2\tau} \end{cases}$$

This can be rewritten as:

$$\begin{cases} 2\varphi E_W^{MP} = \frac{m\tau}{\varphi+m\tau} \\ 2\tau E_D^{MP} = m - \frac{m\tau}{\varphi+m\tau} \end{cases} \quad (\$)$$

Before proceeding the following equalities have to be stated:

$$\begin{cases} E_W^{MP} = \frac{1}{2} \left(\frac{m\bar{x}_0}{\alpha+m}\right)^2 \\ E_D^{MP} = \frac{m}{2} \left[s^2 + \left(\frac{\alpha\bar{x}_0}{\alpha+m}\right)^2\right] \end{cases} \quad (\S)$$

Taking the ratio of equations in (\$) and using (§):

$$\alpha(m + \alpha - 1) = \frac{E_D^{MP}}{E_W^{MP}} = \frac{\alpha^2 \bar{x}_0^2 + s^2(\alpha + m)^2}{m \bar{x}_0^2}$$

That leads to the second order equation:

$$\alpha^2 [(m - 1)\bar{x}_0^2 - s^2] + \alpha \{m [(m - 1)\bar{x}_0^2 - 2s^2]\} - m^2 s^2 = 0$$

Whose solution is:

$$\alpha^{me} = \frac{S^2}{\bar{x}_0^2 - \frac{S^2}{m}}$$

In terms of λ^{me} :

$$\lambda^{me} = \frac{\bar{x}_0^2 - \frac{S^2}{m}}{\bar{x}_0^2}$$

Q.D.E

B

VQSVM section

B.1 Proof of Theorem 3.4.2.1:

Formally the thesis is:

$$(\boldsymbol{\alpha} \ \lambda) \mathbf{H} \begin{pmatrix} \boldsymbol{\alpha} \\ \lambda \end{pmatrix} \geq 0$$

Holds true for every value of $\boldsymbol{\alpha}$ and λ . For convenience of notation \mathbf{Q} is the matrix of elements $q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, and \mathbf{G} the matrix of elements $g_{ij} = y_j \Phi(x_j)^i$

Computing the second derivatives of eq.(3.134) leads to:

$$\begin{pmatrix} \frac{1}{1+\lambda} \mathbf{Q} & \frac{-\mathbf{Q}\boldsymbol{\alpha} + \mathbf{w}^{(KM)} \mathbf{G}}{(1+\lambda)^2} \\ \left(\frac{-\mathbf{Q}\boldsymbol{\alpha} + \mathbf{w}^{(KM)} \mathbf{G}}{(1+\lambda)^2} \right)^t & \frac{\|\mathbf{w}^{(\lambda=0)} - \mathbf{w}^{(KM)}\|^2}{(1+\lambda)^3} \end{pmatrix}$$

By explicit computation one obtains:

$$\begin{aligned} & (\boldsymbol{\alpha} \ \lambda) \mathbf{H} \begin{pmatrix} \boldsymbol{\alpha} \\ \lambda \end{pmatrix} = \\ & = \frac{\boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha}}{1+\lambda} + \frac{\lambda^2 \|\mathbf{w}^{(\lambda=0)} - \mathbf{w}^{(KM)}\|^2}{(1+\lambda)^3} + \frac{2\lambda}{(1+\lambda)^2} \left([\mathbf{w}^{(\lambda=0)}]^t \mathbf{w}^{(KM)} - \boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha} \right) = \\ & = \boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha} / (1 + \lambda) + \frac{\lambda^2}{(1+\lambda)^3} \left(\boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha} - 2 [\mathbf{w}^{(\lambda=0)}]^t \mathbf{w}^{(KM)} + \|\mathbf{w}^{(KM)}\|^2 \right) + \\ & + (2\lambda) / (1 + \lambda)^2 \left([\mathbf{w}^{(\lambda=0)}]^t \mathbf{w}^{(KM)} - \boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha} \right) = \end{aligned}$$

$$\begin{aligned}
 &= \boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha} \left(\frac{1}{1+\lambda} + \frac{\lambda^2}{(1+\lambda)^3} - \frac{2\lambda}{(1+\lambda)^2} \right) + [\mathbf{w}^{(\lambda=0)}]^t \mathbf{w}^{(KM)} \left(\frac{2\lambda}{(1+\lambda)^2} - \frac{2\lambda^2}{(1+\lambda)^3} \right) + \frac{\lambda^2 \|\mathbf{w}^{(KM)}\|^2}{(1+\lambda)^3} = \\
 &= 1/(1+\lambda)^3 \left(\boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha} + 2\lambda [\mathbf{w}^{(\lambda=0)}]^t \mathbf{w}^{(KM)} + \lambda^2 \|\mathbf{w}^{(KM)}\|^2 \right)
 \end{aligned}$$

The term on brackets is $\|\mathbf{w}^{(\lambda=0)} + \lambda \mathbf{w}^{(KM)}\|^2$. Because of the square, and recalling the fact that λ is always positive being a Lagrange multiplier, than the problem is convex. The problem is strictly convex depending on \mathbf{Q} . If \mathbf{Q} is positive definite then the problem is strictly convex, if \mathbf{Q} is only positive definite then the problem is convex. **Q.D.E.**

B.2 Proof of Lemma 3.4.1

In this case it is necessary to compute λ that minimizes (3.136). By computing the derivative along λ and setting to 0 one gets: $\frac{\rho_0^2}{2} - \frac{\|\mathbf{w}^{(\lambda=0)} - \mathbf{w}^{(KM)}\|^2}{2(1+\lambda)^2} - \delta = 0$. From which: $(1 + \lambda)^2 = \frac{\|\mathbf{w}^{(\lambda=0)} - \mathbf{w}^{(KM)}\|^2}{\rho_0^2 - 2\delta}$. When the solution lies outside the bounding sphere the constraint is active and consequently $\lambda > 0$. At the optimum must hold $\lambda\delta = 0$ that leads to $\delta = 0$. In other words the new λ can be estimated by: $(1 + \tilde{\lambda})^2 = \frac{\|\mathbf{w}^{(\lambda=0)} - \mathbf{w}^{(KM)}\|^2}{\rho_0^2}$. Between the two possible solutions, that with $+\sqrt{\frac{\|\mathbf{w}^{(\lambda=0)} - \mathbf{w}^{(KM)}\|^2}{\rho_0^2}}$ must be chosen being $1 + \tilde{\lambda} > 0$ (because $\lambda > 0$).

More explicitly $\tilde{\lambda} = \sqrt{\frac{\boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha} + \|\mathbf{w}^{(KM)}\|^2 - 2[\mathbf{w}^{(\lambda=0)}]^t \mathbf{w}^{(KM)}}{\rho_0^2}} - 1$. Now $\|\mathbf{w}^{(KM)}\|^2 = \sum_{i,j=1}^n \alpha_i^{(KM)} \alpha_j^{(KM)} y_i^{(KM)} y_j^{(KM)} K(\mathbf{x}_i, \mathbf{x}_j)$ and $[\mathbf{w}^{(\lambda=0)}]^t \mathbf{w}^{(KM)} = \sum_{i,j=1}^n \alpha_i \alpha_j^{(KM)} y_i y_j^{(KM)} K(\mathbf{x}_i, \mathbf{x}_j)$. **Q.D.E..**

B.3 Proof of Lemma 3.4.2

Here the objective is to compute D_M when λ is kept fixed. This means that the terms that contain $\boldsymbol{\alpha}$ in D_M are $\left(\frac{\|\mathbf{w}^{(\lambda=0)}\|^2}{2} - \sum_i \alpha_i \right) - \frac{\lambda}{2(1+\lambda)} (\boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha} - 2\boldsymbol{\alpha}^t \boldsymbol{\beta})$ where $\boldsymbol{\beta}$ is the vector of elements $\beta_i = y_i \sum_{j=1}^n \alpha_j^{(KM)} y_j^{(KM)} K(\mathbf{x}_i, \mathbf{x}_j)$. Moreover: $\left(\frac{\boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha}}{2} - \sum_i \alpha_i \right) - \frac{\lambda}{2(1+\lambda)} (\boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha} - 2\boldsymbol{\alpha}^t \boldsymbol{\beta}) = \frac{1}{2(1+\lambda)} \boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha} - \sum_i \alpha_i \left(1 - \frac{\lambda}{(1+\lambda)} \beta_i \right)$ or, multiplying by $(1 + \lambda)$ in a more usual form as:

$$\frac{1}{2} \boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha} - \sum_i \alpha_i (1 + \lambda(1 - \beta_i))$$

B.4 Proof of Lemma 3.4.3

The Lagrange multiplier \hat{b} obtained by the minimization process of (3.140) must be divided by the factor $1 + \lambda$ to get the bias b to be used in the decision function (3.143). This aspect can be simply seen by looking at (3.136) where for obtaining the final cost (3.140) everything was multiplied for $1 + \lambda$ (as for the previous lemma). For this reason, to recover the bias value it is necessary to divide the Lagrange multiplier \hat{b} by $1 + \lambda$.



Biased Regularization section

C.1 Proof of Biased SVM dual theorem

Up to constant terms Biased SVM can be conveniently rewritten as:

$$\left\{ \begin{array}{l} \min_{\varepsilon, \mathbf{w}} C \sum_{i=1}^n \varepsilon_i + \frac{1}{2} \|\mathbf{w}\|^2 - \lambda_2 \mathbf{w} \cdot \mathbf{w}_0 \\ y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 1 - \varepsilon_i \quad \forall i \\ \varepsilon_i \geq 0 \quad \forall i \end{array} \right.$$

The associated Lagrangian is:

$$L(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{w}, \varepsilon) = C \sum_{i=1}^n \varepsilon_i + \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i) - 1 + \varepsilon_i] - \sum_{i=1}^n \beta_i \varepsilon_i - \lambda_2 \mathbf{w} \cdot \mathbf{w}_0$$

The associated derivatives are:

$$\begin{aligned} \nabla_{\mathbf{w}} L(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{w}, \varepsilon) &= \mathbf{w} - \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i - \lambda_2 \mathbf{w}_0 = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i + \lambda_2 \mathbf{w}_0 \\ \frac{\partial L(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{w}, \varepsilon)}{\partial \varepsilon_i} &= C - \alpha_i - \beta_i = 0 \Rightarrow C = \alpha_i + \beta_i \Rightarrow 0 \leq \alpha_i \leq C \quad (\text{because } \alpha_i, \beta_i \geq 0) \end{aligned}$$

Substituting the primal variables $(\mathbf{w}, \varepsilon)$ with the dual $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ one, using the usual notation $q_{ij} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$, recalling that one wants the max of $L(\boldsymbol{\alpha}, \boldsymbol{\beta})$, one gets the final dual problem:

$$\left\{ \begin{array}{l} \min_{\boldsymbol{\alpha}} \frac{1}{2} \boldsymbol{\alpha}^t \mathbf{Q} \boldsymbol{\alpha} - \sum_{i=1}^n \alpha_i (1 - \lambda_2 y_i \mathbf{w}_0^t \mathbf{x}_i) \\ 0 \leq \alpha_i \leq C \quad \forall i \end{array} \right.$$

Only dot products appear on the above formulation. Than one can still use kernels $q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, and the reference function $\mathbf{w}_0^t \mathbf{x}_i$ can be any function $f_0(\mathbf{x}_i)$. Thus:

$$\begin{cases} \min_{\alpha} \frac{1}{2} \alpha^t \mathbf{Q} \alpha - \sum_{i=1}^n \alpha_i (1 - \lambda_2 y_i f_0(\mathbf{x}_i)) \\ 0 \leq \alpha_i \leq C \quad \forall i \end{cases}$$

The final model of the decision function is:

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + \lambda_2 f_0(\mathbf{x})$$

Note that the new function $f(\mathbf{x})$ differs from the original SVM function up to an additive multiple of $f_0(\mathbf{x})$; this is in accordance with the Generalized Representer Theorem in [48].

C.2 Proof of Biased RLS primal solution

Consider the Biased RLS (Tikhonov) primal formulation:

$$\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda_1 \|\mathbf{w} - \lambda_2 \mathbf{w}_0\|^2$$

Or in other terms up to positive constants:

$$\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda_1 \|\mathbf{w}\|^2 - 2\lambda_2 \lambda_1 \mathbf{w} \cdot \mathbf{w}_0$$

Computing the gradient and setting to 0 one gets:

$$2\mathbf{X}^t (\mathbf{X}\mathbf{w} - \mathbf{y}) + 2\lambda_1 \mathbf{w} - 2\lambda_2 \lambda_1 \mathbf{w}_0 = 0$$

Easily it follows that one has to solve the following linear system:

$$(\mathbf{X}^t \mathbf{X} + \lambda_1 \mathbf{I}) \mathbf{w} = \mathbf{X}^t \mathbf{y} + \lambda_2 \lambda_1 \mathbf{w}_0$$

C.3 Proof of Biased RLS dual

One starts from the primal:

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \frac{\lambda_1}{2} \|\mathbf{w} - \lambda_2 \mathbf{w}_0\|^2$$

This can be written in the convenient form:

$$\begin{cases} \min_{\varepsilon, \mathbf{w}} \frac{1}{2} \sum_{i=1}^n \varepsilon_i^2 + \frac{\lambda_1}{2} \|\mathbf{w}\|^2 - \lambda_2 \lambda_1 \mathbf{w} \cdot \mathbf{w}_0 \\ \mathbf{X}\mathbf{w} - \mathbf{y} = \varepsilon \end{cases}$$

The Lagrangian is:

$$L(\mathbf{u}, \varepsilon, \mathbf{w}) = \frac{1}{2} \sum_{i=1}^n \varepsilon_i^2 + \frac{\lambda_1}{2} \|\mathbf{w}\|^2 - \lambda_2 \lambda_1 \mathbf{w} \cdot \mathbf{w}_0 - \mathbf{u}^t (\mathbf{X}\mathbf{w} - \mathbf{y} - \varepsilon)$$

The associated derivatives are:

$$\begin{aligned} \nabla_{\mathbf{w}} L(\mathbf{u}, \mathbf{w}, \varepsilon) &= \lambda_1 \mathbf{w} - \lambda_2 \lambda_1 \mathbf{w}_0 - \mathbf{X}^t \mathbf{u} = 0 \Rightarrow \mathbf{w} = \frac{\mathbf{X}^t \mathbf{u} + \lambda_2 \lambda_1 \mathbf{w}_0}{\lambda_1} \\ \nabla_{\varepsilon} L(\mathbf{u}, \mathbf{w}, \varepsilon) &= \varepsilon + \mathbf{u} = 0 \Rightarrow \varepsilon = -\mathbf{u} \end{aligned}$$

Substituting the primal with the dual variables one gets and after a few algebra one gets:

$$L(\mathbf{u}) = -\frac{1}{2} \mathbf{u}^t \mathbf{u} - \frac{1}{2\lambda_1} \mathbf{u}^t \mathbf{X}\mathbf{X}^t \mathbf{u} - \lambda_2 \mathbf{u}^t \mathbf{X}\mathbf{w}_0 + \mathbf{u}^t \mathbf{y} + \frac{\lambda_1}{2} \|\lambda_2 \mathbf{w}_0\|^2 - \lambda_1 \lambda_2^2 \|\mathbf{w}_0\|^2$$

Recalling that one wants the max of $L(\mathbf{u})$, one can nullify the gradient of $L(\mathbf{u})$ with respect to \mathbf{u} in order to obtain the solution. Then one gets:

$$\nabla_{\mathbf{u}} L(\mathbf{u}) = -\mathbf{u} - \frac{1}{\lambda_1} \mathbf{X}\mathbf{X}^t \mathbf{u} - \lambda_2 \mathbf{X}\mathbf{w}_0 + \mathbf{y} = 0$$

That is one has the following linear system:

$$(\mathbf{X}\mathbf{X}^t + \lambda_1 \mathbf{I}) \frac{\mathbf{u}}{\lambda_1} = \mathbf{y} - \lambda_2 \mathbf{X}\mathbf{w}_0$$

To obtain an usual form one can define a new set of solution variables: $\beta = \mathbf{u}/\lambda_1$. Then one gets $(\mathbf{X}\mathbf{X}^t + \lambda_1 \mathbf{I})\beta = \mathbf{y} - \lambda_2 \mathbf{X}\mathbf{w}_0$. Also in this case only dot

products are involved and the reference function $w_0^t x_i$ can be any function $f_0(x_i)$. Then in general one has:

$$(\mathbf{K} + \lambda_1 \mathbf{I})\boldsymbol{\beta} = \mathbf{y} - \lambda_2 f_0(\mathbf{X}).$$

The final model of the decision function is:

$$f(\mathbf{x}) = \sum_{i=1}^n \beta_i K(\mathbf{x}, \mathbf{x}_i) + \lambda_2 f_0(\mathbf{x})$$

this is in accordance with the Generalized Representer Theorem in [48].



Generalized Tikhonov section

D.1 Proof of Theorem 3.2.1

The expression object of analysis is

$$E_\varepsilon\{\|X(\hat{w} - w_*)\|^2\}$$

As first observation, due to the linearity of $tr()$ and E_ε one has:

$$E_\varepsilon\{\|X(\hat{w} - w_*)\|^2\} = tr[XE_\varepsilon\{(\hat{w} - w_*)(\hat{w} - w_*)^t\}X^t]$$

Another observation is that $\hat{w} - w_* = \hat{w} - E_\varepsilon(\hat{w}) + b$ where b is the bias term. Given this equation one simply shows that:

$$E_\varepsilon\{(\hat{w} - w_*)(\hat{w} - w_*)^t\} = var(\hat{w}) + bb^t$$

Then one obtains that:

$$E_\varepsilon\{\|X(\hat{w} - w_*)\|^2\} = tr[Xvar(\hat{w})X^t] + tr[X(bb^t)X]$$

This last equation using SVD becomes:

$$\begin{aligned} &= \sigma_y^2 tr \left\{ U \Sigma V^t V D_n \left(\frac{\sigma_i^2}{(\sigma_i^2 + \theta_i^2)^2} \right) V^t V \Sigma^t U^t \right\} + \\ &+ tr \left\{ U \Sigma V^t V D_n \left(-\frac{\theta_i^2}{\sigma_i^2 + \theta_i^2} \right) V^t w_* w_*^t V D_n \left(-\frac{\theta_i^2}{\sigma_i^2 + \theta_i^2} \right) V^t V \Sigma^t U^t \right\} \end{aligned}$$

Due to the $tr()$ operator properties one can write:

$$= \sigma_y^2 \text{tr} \left\{ D_n \left(\frac{\sigma_i^2}{(\sigma_i^2 + \theta_i^2)^2} \right) \Sigma^t \Sigma \right\} + \\ + \text{tr} \left\{ V^t w_* w_*^{*t} V D_n \left(-\frac{\theta_i^2}{\sigma_i^2 + \theta_i^2} \right) \Sigma^t \Sigma D_n \left(-\frac{\theta_i^2}{\sigma_i^2 + \theta_i^2} \right) \right\}$$

That can be equivalently rewritten in the final form:

$$\sigma_y^2 \sum_{i=1}^r \frac{\sigma_i^4}{(\sigma_i^2 + \theta_i^2)^2} + \sum_{i=1}^r \frac{\theta_i^4 \sigma_i^2}{(\sigma_i^2 + \theta_i^2)^2} < w_*, v_i >^2$$

Recalling that this equation is the cost then one wants to obtain its minimum. Taking the derivative with respect to each θ_i^2 and setting each equality to 0 one gets:

$$\frac{\theta_i^2 \sigma_i^4}{(\sigma_i^2 + \theta_i^2)^3} < w_*, v_i >^2 = \sigma_y^2 \frac{\sigma_i^4}{(\sigma_i^2 + \theta_i^2)^3} \quad (\text{D.1})$$

To fulfill these equations a sufficient condition is to set each θ_i^2 to the oracular value given by:

$$(\theta_i^{\text{orac}})^2 = \frac{\sigma_y^2}{< w_*, v_i >^2} \quad (\text{D.2})$$

D.2 Proof of 3.64

The expression of the regularized solution in the β vector is

$$\hat{\beta} = \left[D_n \left(\frac{\sigma_i}{\sigma_i^2 + \theta_i^2} \right) \middle| 0_{n,m-n} \right] U^t y = \left[D_n \left(\frac{\sigma_i}{\sigma_i^2 + \theta_i^2} \right) \middle| 0_{n,m-n} \right] U^t (X w_* + \varepsilon) \quad (\text{D.3})$$

that is:

$$\hat{\beta} = \left[D_n \left(\frac{\sigma_i}{\sigma_i^2 + \theta_i^2} \right) \middle| 0_{n,m-n} \right] \Sigma V^t w_* + \left[D_n \left(\frac{\sigma_i}{\sigma_i^2 + \theta_i^2} \right) \middle| 0_{n,m-n} \right] U^t \varepsilon \quad (\text{D.4})$$

Now e_1 , e_2 and e_3 will be separately discussed. For discussing the term e_2 one can note that the generalized Tikhonov cost function can be written as:

$$(y - Xw)^t (y - Xw) + w^t (VT^2V^t)w \quad (\text{D.5})$$

nullifying its gradient one gets the relation:

$$(y - X\hat{w})^t X = \hat{w}^t VT^2V^t \quad (\text{D.6})$$

Using this relation term e_2 can be written as:

$$\begin{aligned} 2E_\varepsilon[(y - X\hat{w})^t X(\hat{w} - w_*)] &= \\ &= 2E_\varepsilon[\hat{w}^t VT^2V^t(\hat{w} - w_*)] = \\ &= 2E_\varepsilon[\hat{\beta}^t T^2 \hat{\beta}] - 2E[\hat{\beta}^t T^2 V^t w_*] = \end{aligned} \quad (\text{D.7})$$

using the expression (D.3) and plugging it in the previous expression and deleting the terms for which $E_\varepsilon[\cdot] = 0$, one has:

$$\begin{aligned} 2E_\varepsilon \left[\varepsilon^t U D_m \left[\left(\frac{\sigma_i \theta_i}{\sigma_i^2 + \theta_i^2} \right)^2 \right] U^t \varepsilon \right] + \\ + 2w^{*t} V D_n \left[\left(\frac{\sigma_i^2 \theta_i}{\sigma_i^2 + \theta_i^2} \right)^2 \right] V^t w_* + \\ - 2w^{*t} V D_n \left(\frac{\sigma_i^2 \theta_i^2}{\sigma_i^2 + \theta_i^2} \right) V^t w_* \end{aligned} \quad (\text{D.8})$$

Using the relation:

$$\left(\frac{\sigma_i^2 \theta_i}{\sigma_i^2 + \theta_i^2} \right)^2 - \frac{\sigma_i^2 \theta_i^2}{\sigma_i^2 + \theta_i^2} = \frac{-\sigma_i^2 \theta_i^4}{(\sigma_i^2 + \theta_i^2)^2} \quad (\text{D.9})$$

Hence one gets:

$$e_2 = 2E_\varepsilon \left[\varepsilon^t U D_m \left[\left(\frac{\sigma_i \theta_i}{\sigma_i^2 + \theta_i^2} \right)^2 \right] U^t \varepsilon \right] - 2w^{*t} V D_n \left[\frac{\sigma_i^2 \theta_i^4}{(\sigma_i^2 + \theta_i^2)^2} \right] V^t w_* \quad (\text{D.10})$$

Concerning e_3 one preliminary observe that:

$$\begin{aligned} \hat{w} - w_* &= \\ &= V \hat{\beta} - w_* = \\ &= U \left[D_n \left(\frac{-\sigma_i \theta_i^2}{\sigma_i^2 + \theta_i^2} \right) \mid 0_{n,m-n} \right] V^t w_* + U D_m \left(\frac{\sigma_i^2}{\sigma_i^2 + \theta_i^2} \right) U^t \varepsilon \end{aligned} \quad (\text{D.11})$$

Using this relation, one gets:

$$E_\varepsilon \left[\varepsilon^t U D_m \left[\left(\frac{\sigma_i^2}{\sigma_i^2 + \theta_i^2} \right)^2 \right] U^t \varepsilon \right] + w^{*t} V D_n \left[\left(\frac{\sigma_i \theta_i^2}{\sigma_i^2 + \theta_i^2} \right)^2 \right] V^t w_* \quad (\text{D.12})$$

Summarizing till now, one has:

$$\begin{aligned} m\sigma_y^2 &= E_\varepsilon [(y - X\hat{w})^t (y - X\hat{w})] + \\ &+ E_\varepsilon \left[\varepsilon^t U D_m \left[1 - \frac{\theta_i^4}{(\sigma_i^2 + \theta_i^2)^2} \right] U^t \varepsilon \right] + \\ &- w^{*t} V D_n \left[\left(\frac{\sigma_i \theta_i^2}{\sigma_i^2 + \theta_i^2} \right)^2 \right] V^t w_* \end{aligned} \quad (\text{D.13})$$

Now the second and third term are re-written: using the relation $\|h^2\| = h^t h = \text{tr}(hh^t)$, the second term of this expression can be written as:

$$\sigma_y^2 \left[r - \sum_{i=1}^r \left(\frac{\theta_i^2}{\sigma_i^2 + \theta_i^2} \right)^2 \right] \quad \text{(D.14)}$$

instead the third term is:

$$\sum_{i=1}^n \frac{\sigma_i^2 \theta_i^4}{(\sigma_i^2 + \theta_i^2)^2} < w_*, v_i >^2 \quad \text{(D.15)}$$

Summarazing all, and setting $\eta_i = \frac{\theta_i^2}{\sigma_i^2 + \theta_i^2}$ one gets:

$$\sigma_y^2 [m - r + \sum_{i=1}^r \eta_i^2] = E_\epsilon [(y - X\hat{w})^t (y - X\hat{w})] - \sum_{i=1}^n \sigma_i^2 \eta_i^2 < w_*, v_i >^2 \quad \text{(D.16)}$$

and the term η_i contains all the information about the generalized regularization given by the matrix T . Solving for σ_y^2 leads to the estimator.

D.3 Proof of kernel matrix eq.(3.71)

One has

$$K = ((\hat{T}X^{-1})^t (\hat{T}X^{-1}))^{-1} \quad \text{(D.17)}$$

then having in mind that $\hat{T} = TV^t$:

$$K = (U\Sigma^{-1}V^tVT^tTV^tV\Sigma^{-1}U^t)^{-1} \quad \text{(D.18)}$$

and:

$$K = (U\Sigma^{-1}T^tT\Sigma^{-1}U^t)^{-1} \quad \text{(D.19)}$$

that leads to:

$$K = (UD_m(\theta_i^2/\sigma_i^2)U^t)^{-1} \quad \text{(D.20)}$$

equivalently due to the orthogonality of U :

$$K = UD_m(\sigma_i^2/\theta_i^2)U^t$$

Bibliography

- [1] B. Schölkopf and A. J. Smola. *Learning with kernels*. MIT Press, 2001.
- [2] M. Kearns and L. G. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, 1989.
- [3] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.
- [4] S. Decherchi M. Parodi S. Ridella. The regularized mean problem. *Submitted*, 2010.
- [5] S. Decherchi M. Parodi S. Ridella. A neural model approach for regularization in the mean estimation case. In *IEEE International Joint Conference on Neural Networks*, 2010.
- [6] S. Ridella S. Decherchi, M. Parodi. On regularization, shrinking, filtering and learning. *In preparation*, 2010.
- [7] J. Weston R. Collobert F. Sinz L. Bottou V. Vapnik. Inference with the universum. In W. W. Cohen A. Moore, editor, *Proceedings of the 23rd International Conference on Machine Learning*, pages 1009–1016. ACM Press, New York, NY, USA, 2006.
- [8] S. Decherchi S. Ridella R. Zunino P. Gastaldo and D. Anguita. Using unsupervised analysis to constrain generalization bounds for support vector classifiers. *IEEE Transactions on Neural Networks*, 2010.
- [9] S. Decherchi P. Gastaldo J. Redi R. Zunino. Maximal-discrepancy bounds for regularized classifiers. In *IEEE International Joint Conference on Neural Networks*, 2009.
- [10] S. Decherchi P. Gastaldo R. Zunino. Biased regularization for semi-supervised classification. *Submitted*, 2010.
- [11] S. Decherchi S. Ridella P. Gastaldo R. Zunino. Explicit overall risk minimization transductive bound. In *ICNPAA*, 2008.

- [12] S. Decherchi P. Gastaldo M. Parodi R. Zunino. Low complexity linear circuit implementation of support vector machine training. *Electronics Letters*, 2008.
- [13] S. Decherchi P. Gastaldo M. Parodi R. Zunino. Circuit implementation of svm training. In *IEEE International Joint Conference on Neural Networks*, 2009.
- [14] Sergio Decherchi, Paolo Gastaldo, and Rodolfo Zunino. Efficient approximate regularized least squares by toeplitz matrix. *Pattern Recognition Letters*, 32(3):468 – 475, 2011.
- [15] S. Decherchi P. Gastaldo F. Sangiacomo A. Leoncini R. Zunino. Operative assessment of predicted generalization errors on non-stationary distributions in data-intensive applications. *Intelligent Data Analysis*, (in press), 2010.
- [16] S. Decherchi P. Gastaldo R. Zunino. Non-stationary data mining: the network security issue. In *ICANN*, 2008.
- [17] S. Decherchi P. Gastaldo R. Zunino. *Advances In Artificial Intelligence for Privacy Protection and Security*, chapter K-Means clustering for Content Based Document Management in Intelligence. World Scientific Publishing, 2009.
- [18] F. Sangiacomo A. Leoncini S. Decherchi P. Gastaldo R. Zunino. Sealab advanced information retrieval. In *IEEE Int. Conf. Semantic Computing ICSC*, 2010.
- [19] S. Decherchi P. Gastaldo J. Redi R. Zunino. Hypermetric k-means clustering for content-based document management. In *CISIS*, 2008.
- [20] S. Decherchi S. Tacconi J. Redi A. Leoncini F. Sangiacomo and R. Zunino. Text clustering for digital forensics analysis. In *CISIS*, 2009.
- [21] S. Decherchi P. Gastaldo J. Redi R. Zunino. A text clustering framework for information retrieval. *Journal of Information Assurance and Security, Special Issue CISIS 2008*, 2009.

- [22] S. Decherchi S.Tacconi J.Redì F. Sangiacomo A. Leoncini R. Zunino. Text clustering for digital forensics analysis. *Journal of Information Assurance and Security, Special Issue CISIS 2009*, 2010.
- [23] S. Decherchi P. Gastaldo and R. Zunino. Regularization strategies for the extreme learning machine. *Submitted*.
- [24] D. Leoncini S. Decherchi O. Faggioni P. Gastaldo M. Soldani and R. Zunino. A preliminary study on svm based analysis of underwater magnetic signals for port protection. In *CISIS*, 2009.
- [25] D. Leoncini S. Decherchi O.Faggioni P. Gastaldo M. Soldani R. Zunino. Linear svm for underwater magnetic signals based port protection. *Journal of Information Assurance and Security, Special Issue CISIS 2009*, 2010.
- [26] P. L. Bartlett S. Boucheron and G. Lugosi. Model selection and error estimation. *Machine Learning*, 48:85–113, 2002.
- [27] D. A. McAllester. Some pac-bayesian theorems. *Machine Learning*, 37:355–363, 1999.
- [28] P. L. Bartlett S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3, 2002. <http://jmlr.csail.mit.edu/papers/volume3/bartlett02a/bartlett02a.pdf>.
- [29] O. Bousquet A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.
- [30] A. Cannon J.M. Ettinger D. Hush C. Scovel. Machine learning with data dependent hypothesis classes. *Journal of Machine Learning Research*, 2:335–358, 2002.
- [31] J. Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, 6:273–306, 2005.

- [32] T. Poggio and F. Girosi. A theory of networks for approximation and learning. *A.I. Memo No.1140, C.B.I.P. Paper No. 31*, 1989.
- [33] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386-408, 1958.
- [34] G. Cybenko. Approximations by superpositions of sigmoidal functions. *Mathematics of Control, Signals, and Systems*, pages 303–314, 1989. http://actcomm.dartmouth.edu/gvc/papers/approx_by_superposition.pdf.
- [35] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4, 1991.
- [36] D. E. Rumelhart G. E. Hinton R. J. Williams. Learning representations by back-propagating errors. *Nature*, 1986.
- [37] M.T. Hagan M.B. Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5:989–993, 1994.
- [38] P.L. Bartlett. For valid generalization, the size of the weights is more important than the size of the network. In *Advances in Neural Information Processing Systems*, pages 134–140. The MIT Press, 1997.
- [39] A. N. Tikhonov V. Y. Arsenin. *Solution of Ill-posed Problems*. Winston Sons, 1977.
- [40] L. Rosasco A. Caponnetto E. De Vito U. De Giovannini F. Odone. Learning, regularization and ill-posed inverse problems. In *Advances in Neural Processing Systems*, 2004.
- [41] T. Poggio T. Evgeniou, M. Potil. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 1999.
- [42] H. Hochstadt. *Integral Equations*. John Wiley Sons, 1973.

- [43] J. Stewart. Positive definite functions and generalizations, an historical survey. *Rocky Mountain J. Math.*, 6:409-434, 1976.
- [44] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 1998.
- [45] F. Steinke and B. Schölkopf. Kernels, regularization and differential equations. *Pattern Recognition*, 41(11):3271–3286, 2008.
- [46] L. Lo Gerfo L. Rosasco F. Odone E. De Vito A. Verri. Spectral algorithms for supervised learning. *Neural Computation*, 20(7):1873–1897, 2008. <http://www.mitpressjournals.org/doi/pdf/10.1162/neco.2008.05-07-517>.
- [47] G. Wahba. Spline models for observational data. In *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, volume 59, 1990.
- [48] B. Schölkopf R. Herbrich A. Smola. A generalized representer theorem. In David Helmbold and Bob Williamson, editors, *Computational Learning Theory*, volume 2111 of *Lecture Notes in Computer Science*, pages 416–426. Springer Berlin / Heidelberg, 2001.
- [49] S. Thrun. Is learning the n-th thing any easier than learning the first? In *Advances in Neural Information Processing Systems*, volume 8, pages 640–646. MIT Press, 1996.
- [50] K. Fukumizu A. Gretton X. Sun B. Scholkopf. Kernel measures of conditional dependence. In *Advances In Neural Processing Systems*, 2007.
- [51] A. Rakotomamonjy F.R. Bach S.Canu Y. GrandValet. Simplemkl. *Journal of Machine Learning Research*, pages 2491–2521, 2008.
- [52] C. S. Ong X. Mary S. Canu and A. J. Smola. Learning with non-positive kernels. In *International Conference on Machine Learning*, page 639-646, 2004.

- [53] R Rifkin G Yeo T Poggio. Regularized least squares classification. In *Advances in Learning Theory: Methods, Model and Applications. NATO Science Series III: Computer and Systems Sciences*, volume 190. IOS Press, 2003.
- [54] S. Keerthi K. Duan S. Shevade A. Poo. A fast dual algorithm for kernel logistic regression. *Machine Learning*, 61:151–165, 2005. 10.1007/s10994-005-0768-5.
- [55] M. R. Hestenes E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49, 1952. <http://nvl.nist.gov/pub/nistpubs/jres/049/6/V49.N06.A08.pdf>.
- [56] R. M. Rifkin. Everything old is new again: A fresh look at historical approaches in machine learning. 2002. PhD thesis.
- [57] C. Burges. Simplified support vector decision rules. In L. Sarta, editor, *13th International Conference on Machine Learning*, page 7177, 1996.
- [58] S. Boyd L. Vanderberghe. *Convex Optimization*. Cambridge University Press.
- [59] N. Cristianini J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning*. Cambridge University Press, 2000.
- [60] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in kernel methods: Support vector machines*. MIT Press, 1998.
- [61] S. Keerth S. K. Shevade C. Bhattacharyya K. R. K. Murthy. Improvements to platt’s smo algorithm for svm classifier design. *Neural Computation*, 13(3):637–649, 2001.
- [62] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

- [63] T. Poggio S. Mukherjee R. Rifkin A. Rakhlin. b. *AI Memo 2001-011 CBCL Memo 198*, 2001.
- [64] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, 58:267–288, 1996.
- [65] J. S. Taylor D. Hadoon. Pac-bayes analysis of maximum entropy classification. In *AISTATS 2009*, volume 5, 2009.
- [66] S. Ridella S. Rovetta R. Zunino. Circular backpropagation networks for classification. *IEEE transactions on neural networks*, 8:84–97, 1997.
- [67] A.M. Turing. Intelligent machinery. *Machine Intelligence*, 5:3–23, 1969.
- [68] D. S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Syst.*, 2:321355, 1988.
- [69] B. Igelnik and Yoh-Han Pao. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE J_NN*, 6(6):1320–1329, 1995.
- [70] G.B. Huang Q.Y. Zhu C.K. Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70:489501, 2006.
- [71] L. P. Wang and C. R. Wan. Comments on the extreme learning machine. *IEEE_J_NN*, 19(8) : 1494 – –1495, 2008.
- [72] T. P. Vogl J. K. Mangis A. K. Rigler W. T. Zink and D. L. Alkon. Accelerating the convergence of the back-propagation method. *Biological Cybernetics*, 59:257–263, 1988.
- [73] G.B. Huang L. Chen and C.K. Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, 17:879–892, 2006.
- [74] M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Trans. Neural Netw.*, 13:780–784, 2002.

- [75] W.Y. Chen Y. Song H. Bai C.J. Lin E. Y. Chang. Parallel spectral clustering in distributed systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.
- [76] S. Ridella S. Rovetta and R. Zunino. Plastic algorithm for adaptive vector quantization. *Neural Computing and Applications*, 7(1):37–51, 1998.
- [77] S. Ridella S. Rovetta R. Zunino. K-winner machines for pattern classification. *IEEE Trans Neural Netw*, 12:371–385, 2001.
- [78] S.P. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inf. Theory*, 28:129–135, 1982.
- [79] I. Dhillon, Y. Guan, and B. Kulis. A unified view of kernel k-means, spectral clustering and graph cuts. Technical report, 2004.
- [80] S. Dasgupta and A. Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures and Algorithms*, 22:6065, 2003.
- [81] D. Achlioptas. Database-friendly random projections. In *Symposium on Principles of Database Systems (PODS)*, pages 274–281, 2001.
- [82] R. Rifkin R.A. Lippert. Notes on regularized least-squares. *CBCL Paper 268/AI Technical Report 2007-019, Massachusetts Institute of Technology, Cambridge, MA,, 2007.*
- [83] B. Efron R. Tibshirani. *An Introduction to the Bootstrap*. Chapman Hall/CRC, 1994.
- [84] D. Anguita S. Ridella F. Riviuccio R. Zunino. Automatic hyperparameter tuning for support vector machines. In José Dorronsoro, editor, *Artificial Neural Networks ICANN 2002*, volume 2415 of *Lecture Notes in Computer Science*, pages 83–83. Springer Berlin / Heidelberg, 2002.
- [85] W. James C. Stein. Estimation with quadratic loss. In *Proc. Fourth Berkeley Symp. Math. Statist. Prob.*, volume 1, page 361379, 1961.

- [86] J.E. Moody. The effective number of parameters: An analysis of generalisation and regularisation in nonlinear learning systems. In *Neural Information Processing Systems*, volume 4, pages 847–854, 1992.
- [87] S. Geisser. *Predictive Inference*. Chapman and Hall, 1993.
- [88] D.J.C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4:448–472.
- [89] J.R. Thompson. Some shrinkage techniques for estimating the mean. *Journal of the American Statistical Association*, 63:113–122, 1968.
- [90] J. B. Copas. Regression prediction and shrinkage. *Journal of the Royal Statistical Society. Series B*, 45(3), 1983.
- [91] F. H. Sinz O. Chapelle A. Agarwal B. Schölkopf. An analysis of inference with the universum. In J. C. Platt D. Koller Y. Singer S. Roweis, editor, *Advances in Neural Information Processing Systems 20: 21st Annual Conference on Neural Information Processing Systems*, pages 1369–1376. Curran, Red Hook, NY, USA, 2007.
- [92] C. Stein. Inadmissibility of the usual estimator for the mean of a multivariate distribution. In *Proc. Third Berkeley Symp. Math. Statist. Prob.*, volume 1, page 197206, 1956.
- [93] P. W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics: Theory and Methods*, page 813827, 1977.
- [94] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions On Information Theory*, 51(12):4203–4215, 2005.
- [95] R. Kohavi. A study of cross validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, 1995.
- [96] O. Chapelle B. Scholkopf A. Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006. <http://www.kyb.tuebingen.mpg.de/ssl-book>.

- [97] X. Zhu. Semi-supervised learning literature survey. *Technical Report 1530, 2008 University of WisconsinMadison*. http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf.
- [98] N.B. Karayiannis MI G. Weiqun. Growing radial basis neural networks: merging supervised and unsupervised learning with network growth techniques. *IEEE Trans. Neural Netw.*, 8:1492–1506, 1997.
- [99] Y. LeCun L. Bottou Y. Bengio and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86:2278–2324, 1998. <http://yann.lecun.com/exdb/mnist/>.
- [100] S. Hettich and S. D. Bay. The uci kdd archive. 1999. <http://kdd.ics.uci.edu>.
- [101] Thorsten Joachims. Making large-scale support vector machine learning practical. *Advances in kernel methods: support vector learning*, pages 169–184, 1999.
- [102] M. F. Porter. An algorithm for suffix stripping. 14:130–137, 1980.
- [103] G. Salton A. Wong and L.S. Yang. A vector space model for information retrieval. *Journal Amer. Soc. Inform. Sci.*, 18:613–620.
- [104] W. Kienzle K. Chellapilla. Personalized handwriting recognition via biased regularization. In *Proceedings of the 23 rd International Conference on Machine Learning*, 2006.
- [105] S. Ben-David T. Lu D. Pál. Does unlabeled data provably help? worst-case analysis of the sample complexity of semi-supervised learning. In *COLT*, 2008.
- [106] A. Ambroladze E. Parrado-Hernandez J. Shawe-Taylor. Tighter pac-bayes bounds. In *Advances in Neural Information Processing Systems*, volume 19, pages 9–16. MIT Press, 2007.
- [107] M. Belkin P. Niyogi and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, pages 2399–2434, 2006.

- [108] R.E. Fan K.W. Chang C.J. Hsieh X.-R. Wang C.J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.
- [109] D. Erhan Y. Bengio A. Courville P.A. Manzagol P. Vincent S. Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11:625–660, 2010.
- [110] P. E. Utgoff. Shift of bias for inductive concept learning. 1984.
- [111] T.M. Mitchell. The need for biases in learning generalizations. 1980. CBM-TR 5-110, Rutgers University, New Brunswick, NJ.
- [112] K. Nigam A. McCallum and T. Mitchell. *Semi-supervised Text Classification Using EM*. MIT Press, 2006.
- [113] O. Chapelle J. Weston and B. Schölkopf. Cluster kernels for semi-supervised learning. In *Advances in Neural Information Processing Systems*, volume 15, pages 585–592. MIT Press, 2003.
- [114] Z. Guo Z. Zhang E. P. Xing and C. Faloutsos. Semi-supervised learning based on semiparametric regularization. In *SIAM Conference on Data Mining*, 2008.
- [115] A. Blum T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, pages 92–100, 1998.
- [116] R. Collobert F. Sinz J. Weston L. Bottou. Large scale transductive svms. *Journal Machine Learning Research*, pages 1687–1712, 2006.
- [117] http://manifold.cs.uchicago.edu/manifold_regularization/manifold.html.
- [118] Y. Le Cun B. Boser J. S. Denker D. Henderson R. E. Howard W. Hubbard and L. J. Jackel. Handwritten digit recognition with a backpropagation network. In *Advances in Neural Information Processing Systems*, volume 2, pages 396–404, 1990.

- [119] <http://www.mathworks.com/matlabcentral/fileexchange/8636-emgm>.
- [120] Don Hush and Clint Scovel. Concentration of the hypergeometric distribution. *Statistics Probability Letters*, 75(2):127 – 132, 2005.
- [121] C. Cortes M. Mohri. On transductive regression. In *NIPS*, volume 19, page 305312, 2007.
- [122] Ran El-Yaniv and Dmitry Pechyony. Transductive rademacher complexity and its applications. In Nader Bshouty and Claudio Gentile, editors, *Learning Theory*, volume 4539 of *Lecture Notes in Computer Science*, pages 157–171. Springer Berlin / Heidelberg, 2007.
- [123] H. Chen W. Chung J.J. Xu G. Wang Y. Qin M. Chau. Crime data mining: a general framework and some examples. *IEEE Trans. Computer*, 37(4):50–56, 2004.
- [124] Hsinchun Chen and Fei-Yue Wang. Artificial intelligence for homeland security. *IEEE Trans. Intelligent Systems*, 20(5), 2005.
- [125] J. W. Seifert. Data mining and homeland security: An overview. *CRS Report RL31798*, 2007.
- [126] J. Mena. Investigative data mining for security and criminal detection. 2003.
- [127] B. Schneier. Why data mining wont stop terror.
- [128] K.A. Taipale. Data mining and domestic security: Connecting the dots to make sense of data. *The Columbia Science And Technology Law Review*, 5:1–83, 2003.
- [129] G. Petasis V. Karkaletsis G. Paliouras I. Androutsopoulos and C.D. Spyropoulos. A new text engineering platform. In *Proc. 3rd International Conference on Language Resources and Evaluation*, 2002.

- [130] D. Ferrucci and A. Lally. Uima: An architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348, 2004.
- [131] F. Sebastiani. *Text Categorization*. WIT Press, 2005.
- [132] D. Sullivan. *Document warehousing and text mining*. John Wiley and Sons, 2001.
- [133] W. Fan L. Wallace S. Rich Z. Zhang. Tapping the power of text mining. *Communications of the ACM*, 49(9):7682, 2006.
- [134] R. Popp T. Armour T. Senator and K. Numrych. Countering terrorism through information technology. *Comm. of the ACM*, 47(3):36–43, 2004.
- [135] B. Goertzel J. Venuto. Accurate svm text classification for highly skewed data using threshold tuning and query-expansion-based feature selection. In *Proc. International Joint Conference on Neural Networks*, pages 1220–1225, 2006.
- [136] N.L. Beebe and G. Dietrich. Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results. *Digital Investigation*, 4:49–54, 2007.
- [137] A. Zanasi. *Text Mining and its Applications to Intelligence*. WIT Press, 2007.
- [138] C. D. Manning P. Raghavan and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [139] D. D. Lewis and K. Sparck Jones. Natural language processing for information retrieval. *Comm. ACM*, 29:99–101.
- [140] R. Baeza-Yates and B. Ribiero-Neto. *Modern Information Retrieval*. ACM Press, New York, 1999.
- [141] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523, 1988.

- [142] G. Salton G. and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.
- [143] F. Shahnaz M. W. Berry V. Paul Pauca R. J. Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing and Management*, 42:373386, 2006.
- [144] G. Qian S. Sural Y. Gu S. Pramanik. Similarity between euclidean and cosine angle distance for nearest neighbor queries. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 1232–1237, 2004.
- [145] Y.-J. Horng S.-M. Chen Y.-C. Chang and C.-H. Lee. A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques. *IEEE Transactions on fuzzy systems*, 13, 2005.
- [146] W. C. Tjhi L. Chen. A heuristic-based fuzzy co-clustering algorithm for categorization of high-dimensional data. *Fuzzy Sets and Systems*, 159:371 389, 2008.
- [147] K. M. Hammouda and M. S. Kamel. Efficient phrase-based document indexing for web document clustering. *IEEE Transactions on Knowledge and Data Engineering*, 16:12791296, 2004.
- [148] H. Chim X. Deng. Efficient phrase-based document similarity for clustering. *IEEE transaction on knowledge and data engineering*, 3(6), 2007.
- [149] O. Zamir and O. Etzioni. Grouper: A dynamic clustering interface to web search results. *Computer Networks*, 31:1361–1374, 1999.
- [150] B. Tang M. Shepherd E. Milios M. Heywood. Comparing and combining dimension reduction techniques for efficient text clustering. In *International Workshop on Feature Selection for Data Mining*, 2005.

- [151] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. *Knowledge Discovery and Data Mining*, page 269274, 2001.
- [152] D. Hand H. Mannila P. Smyth. *Principles of Data Mining*. MIT Press Cambridge MA.
- [153] M.W. Berry S.T. Dumais G. W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573595, 1995.
- [154] D. Cai X. He and J. Han. Document clustering using locality preserving indexing. *IEEE Transaction on knowledge and data engineering*, 17, 2005.
- [155] F. Shahnaz M. W. Berry V. Paul Pauca R. J. Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing and Management*, 42:373386, 2006.
- [156] G. Biau L. Devroye and G. Lugosi. On the performance of clustering in hilbert spaces. *IEEE Trans. on Information Theory*, 54, 2008.
- [157] The Enron Email Dataset.
- [158] L. Jing M. K. Ng and J. Zhexue Huang. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Transactions on knowledge and data engineering*, 19:1026–1041, 2007.
- [159] L.O. Chua and G.N. Lin. Non linear programming without computation. *IEEE Trans. on Circuits and Systems*, 31:182–188, 1984.
- [160] D. Anguita S. Ridella S. Rovetta. Circuitual implementation of support vector machines. *Electronics Letters*, 34:1596–1597, 1998.
- [161] L.O. Chua C. A. Desoer and E. S. Kuh. *Linear and Nonlinear Circuits*. 1987.

- [162] S. Boughorbel J. Tarel F. Fleuret N. Boujemaa. Gcs kernel for svm-based image recognition. In *Int. Conf. on Artificial Neural Networks, ICANN*, volume 2, pages 595–600, 2005.
- [163] W. H. Press B. P. Flannery S. A. Teukolsky and W. T. Vetterling. *Numerical Recipes in FORTRAN: The Art of Scientific Computing, 2nd ed.* Cambridge, England: Cambridge University Press, 1992.
- [164] B. R. Musicus. Levinson and fast cholesky algorithms for toeplitz and almost toeplitz matrices. *RLE TR No. 538, MIT*, 1988.
- [165] L. Rabiner B.H. Juang. *Fundamentals of speech recognition*. Prentice Hall, 1993.
- [166] S. Decherchi G. Parodi P. Gastaldo R. Zunino. Embedded electronics systems for training support vector machines. In *Proc. Int. Joint Conf. Neural Networks IJCNN '06*, pages 2838–2844, 2006.
- [167] S.S Al-Homidan. Sqp algorithms for solving toeplitz matrix approximation problem. *Numerical Linear Algebra with Applications*, 9:619–627, 2002.
- [168] T. F. Chan and P. C. Hansen. A look-ahead levinson algorithm for general toeplitz systems. *IEEEJSP*, 40(5) : 1079 – 1090, 1992.
- [169] S. S. Keerthi C. J. Lin. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural Computation*, 5:1667–1689, 2003.
- [170] S. Munder and D. M. Gavrilu. An experimental study on pedestrian classification. *IEEEJPAAMI*, 28(11) : 1863 – 1868, 2006.
- [171] CBCL. Face database 1, mit center for biological and computation learning. <http://www.ai.mit.edu/projects/cbcl>.
- [172] B. Catanzaro N. Sundaram K. Keutzer. Fast support vector machine training and classification on graphics processors. In *Proceedings of the 25th international conference on Machine learning*, pages 104–111, 2008. <http://www.cs.berkeley.edu/~catanzar/GPUSVM/>.

- [173] S. S. Keerthi S. K. Shevade. Smo algorithm for least-squares svm formulation. *Neural Computation*, 15:487–507, 2003.
- [174] G.B. Huang X.Ding H.Zhou. Optimization method based extreme learning machine for classification. *Neurocomputing*, 2010.
- [175] W. Deng Q. Zheng L. Chen. Regularized extreme learning machine. In *CIDM*, pages 389–395, 2009.
- [176] X. Tang M. Han. Partial lanczos extreme learning machine for single-output regression problems. *Neurocomputing*, 72:30663076, 2009.
- [177] Z.Shi Q.Liu, Q.He. Extreme support vector machine classifier. volume 5012 of *Lecture Notes in Computer Science*, page 222233. 2008.
- [178] B.Frenay M.Verleysen. Using svms with randomized featurespaces :an extreme learning approach. In *ESANN*, page 315320, 2010.
- [179] Luis Torgo. <http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html>.
- [180] B. Mirkin. *Clustering for Data Mining: a Data-recovery Approach*. 2006.
- [181] K. Duan S. Keerthi A. Poo. Evaluation of simple performance measures for tuning svm hyperparameters. Technical report, 2001.
- [182] S. Floyd M. Warmuth. Sample compression, learnability, and the vapnik-chervonenkis dimension. *Machine Learning*, 21:1–36, 1995.
- [183] RC. Williamson J. Shawe-Taylor B. Schölkopf AJ. Smola. Sample based generalization bounds. In *NeuroCOLT2 Tech. Rep. Series, NC-TR-1999-055*, 1999.
- [184] M . Anthony N. Biggs. *Computational Learning Theory*. Cambridge Univ. Press, 1992.
- [185] C. Alippi M. Roveri. Just-in-time adaptive classifiers - part i: Detecting nonstationary changes. *IEEE Tansactions On Neural Networks*, 19(7), 2008.

- [186] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.
- [187] W. N. Street and Y. Kim. A streaming ensemble algorithm (sea) for large-scale classification. *Knowledge Discovery Data Mining*, pages 377–382, 2001.
- [188] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33, 1962.
- [189] Y. Muto H. Nagase and Y. Hamamoto. Evaluation of a modified parzen classifier in high-dimensional spaces. In *Proc. 15th Int’l Conf. Pattern Recognition*, volume 2, pages 67–70, 2000.
- [190] A.P. Dempster N.M. Laird D.B. Rubin. Maximum-likelihood from incomplete data via the em algorithm. *J. Royal Statist. Soc. Ser. B.*, 39, 1977.
- [191] C. Archambeau J.A. Lee and M. Verleysen. On the convergence problems of the em algorithm for finite gaussian mixtures. In *ESANN*, page 99106, 2003.
- [192] S. Ridella R. Zunino. Empirical measure of multiclass generalization performance: the k-winner machine case. *IEEE Trans. Neural Networks*, 12(6):15251529, 2001.
- [193] Kdd cup 1999 intrusion detection dataset:.
- [194] Spamassassin apache project, <http://spamassassin.apache.org/>.
- [195] S. Munder D.M. Gavrilu. An experimental study on pedestrian classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1863–1868, 2006.
- [196] N. Dalvi P. Domingos Mausam S. Sanghai D. Verma. Adversarial classification. In *Proc. KDD’04*, 2004.

- [197] D. Anguita S. Ridella F. Riviaccio R. Zunino. Hyperparameter tuning criteria for support vector classifiers. *Neurocomputing*, 55:109–134, 2003.
- [198] I. J. Taneja and P. Kumar. Relative information of type s, csiszars f-divergence, and information inequalities. *Information Sciences*, 166:105125, 2004.
- [199] I. Csiszar. Information-type distance measures and indirect observations. *Stud. Sci. Math. Hungar*, 2:299–318, 1967.
- [200] W. Lee S.J. Stolfo. Data mining approaches for intrusion detection. 1998.
- [201] S. Ridella R.Zunino. Using k-winner machines for domain analysis. 62:367–388, 2004.
- [202] S. Ridella S. Rovetta R. Zunino. Circular backpropagation networks embed vector quantization. *IEEE Trans. Neural Networks*, 10(4):972975, 1999.
- [203] I. Katakis G. Tsoumakas E. Banos N. Bassiliades I. Vlahavas. An adaptive personalized news dissemination system. *Journal of Intelligent Information Systems*, 32(2):191–201, 2009.
- [204] M.R. De Yao Azimi-Sadjadi A.A. Jamshidi G.J. Dobeck. A study of effects of sonar bandwidth for underwater target classification. *IEEE Journal of Oceanic Engineering*, 27(3):619 – 627, 2002.
- [205] D. Li M. R. Azimi-Sadjadi and M. Robinson. Comparison of different classification algorithms for underwater target discrimination. *IEEE Transactions on Neural Networks*, 15(1):189–194, 2004.
- [206] M. R. Azimi-Sadjadi De Yao Q. Huang G.J. Dobeck. Underwater target classification using wavelet packets and neural networks. *IEEE Transactions on Neural Networks*, 11(3):784–794, 2000.
- [207] M. R. Azimi-Sadjadi De Yao A. Arta Jamshidi and J.G. Dobeck. Underwater target classification in changing environments using an adaptive

- feature mapping. *IEEE Transactions on Neural Networks*, 13(5):1099–1111, 2002.
- [208] R. P. Gorman and T. J. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1:75–89, 1988.
- [209] R.J. Urick. *Principles of Underwater Sound*. McGraw-Hill (New York), 1983.
- [210] O. Faggioni A. Gabellone R. Hollett R.T. Kessel M. Soldani. Anti-intruder port protection mac (magnetic acoustic) system: advances in the magnetic component performance. In *1st WSS Conference*, 2008.
- [211] A. Gabellone O. Faggioni M. Soldani P. Guerrini. In *CAIMAN (Coastal Anti Intruder MAagnetometers Network)*, *Proceedings of RTO-MP-SET-130 Symposium on NATO Military 26 Sensing, NATO classified*, 2008.