# Maximal-Discrepancy Bounds for Regularized Classifiers

Sergio Decherchi, Paolo Gastaldo, Judith Redi, Rodolfo Zunino

*Abstract* — **Regularized Classifiers such as SVM or RLS are among the most used and successful classifiers in machine learning. The theory and the empirical evaluation of the associate generalization bounds are of paramount importance; bounds based on the Maximal-Discrepancy approach proved quite effective. The paper shows an efficient, iterative procedure to evaluate Maximal-Discrepancy bounds for this kind of classifiers. Empirical results on UCI datasets show that this approach can attain tighter bounds to the run-time classification error.**

## I. Introduction

A crucial issue in designing and tuning a classifier is the assessment of the generalization error for a given classification task. A good estimation of the generalization error gives a twofold benefit: first, it guarantees the run-time classification performance of a system; secondly, one can use the predicted error value to perform model selection (parameters tuning). Two approaches exist to deal with theoretical generalization error estimation: a data-independent, worst-case-based one [1], and a second one, strictly dependent on the analyzed problem, providing tighter bounds [2]. In both cases, bounds estimation relies on the notion of Complexity [1][2].

In the present work, a method to tighten data-dependent generalization bounds for Regularized Classifiers is proposed. In particular, the paper focuses on the second class of bounds [2], and the Maximal Discrepancy framework. Regularized classifiers try to minimize the generalization error constraining the space of possible models to a smaller subspace, adopting a complexity minimization strategy [3]: in this context, the interpretation of Complexity is straightforward; indeed, it is identifiable with the regularization term of the classifier cost function. This particular feature can be successfully exploited to obtain a more precise estimation of the generalization bounds.

The present work relies on the observation that most regularized machines use Tickhonov-like regularization terms [3] in the cost function, and therefore it is not possible to set a precise and locked bound to that regularization term. Such a bound, though, can be explicitly built by adding a quadratic constraint to the minimization problem. By properly adding the new defined constraint, the modified problem can fully replace the original one, also guaranteeing the bounding of the regularization term, and therefore avoiding the complexity explosion.

The authors are with University of Genoa, Department of Biophysical and Electronic Engineering, via Opera Pia 11/a, 16145 Genova, Italy, Phone +39 010 3532269 (e-mail {sergio.decherchi, paolo.gastaldo, judith.redi, rodolfo.zunino} @unige.it)

To validate the proposed method, two of the most widely used regularized classifiers, the Support Vector Machine (SVM) and the Regularized Least Squares Method (RLS), are trained on three well-known UCI datasets (Pima-Indians Diabetes, Ionosphere and Sonar).

The remainder of the paper is organized as follows: in section 2, an overview of common regularized classifiers is given; generalization bounds are discussed in section 3, and the proposed approach is outlined in section 4. Finally, section 5 reports the experimental validation of the method.

For brevity, in the continuation of the paper, we will refer to SVM or RLS indistinctly as Regularized Classifiers (RC).

## II. Regularized Classifiers

Classification methods typically aim at minimizing the misclassification error with respect to the whole population, namely the empirical risk, $R_{emp}[f]$. In addition to that goal, regularization methods [3][4], aim to restrict the set of possible models in the hypothesis space, $\Lambda$, to a certain subspace: this approach can be also defined a complexity minimizing strategy [3].

The tradeoff between the empirical risk and the regularization term is usually ruled by a positive constant term, $C$. The cost to be minimized can be expressed in a formal manner as:

$$R_{reg} = C\, R_{emp}[f] + \Omega[f] \tag{1}$$

where the regularization operator $\Omega[f]$ quantifies the complexity of the class of functions from which $f$ is drawn.

When dealing with regularized classifier, $\Omega[f]$ is a square norm $\| f \|_K^2$ in a Reproducing Kernel Hilbert Space $K$, and $K(\mathbf{x}_i, \mathbf{x}_j)$ denotes the dot product in the feature space induced by a kernel function $K(\cdot,\cdot)$. The Representer Theorem [3] proves that the solution of the regularized risk (1) can be expressed as a finite summation, over a set $Z=\{(\mathbf{x}_l, y_l);\ l=1,..,\ n_p\ ;\ y_l \in \{-1,+1\}\}$ of $n_p$ training patterns, as:

$$f(\mathbf{x}_j) = \sum_{i=1}^{n_p} \beta_i K(\mathbf{x}_i, \mathbf{x}_j) \tag{2}$$

Support Vector Machines and Regularized Least Squares Methods are popular methods belonging to this family of regularizing algorithms, and provide excellent performances in pattern recognition problems. The main difference between the two approaches lies in the evaluation of the empirical risk. The two learning algorithms differ in the loss function and in the presence of a bias term. The SVM model

978-1-4244-3553-1/09/$25.00 ©2009 IEEE

uses the 'hinge' loss function, whereas RLS operates on a square loss function. From a mathematical viewpoint, the SVM training process leads to the following optimization problem:

$$\min_f C\sum_{i=1}^{n_p}(1-y_i f_i)_+ + \frac{1}{2}\|f\|_K^2 \quad (3)$$

Problem (2) can be efficiently solved in its dual form by using quadratic programming techniques (e.g. SMO) [5]. In particular, when using the dual formulation of (3), one optimizes a set of Lagrange multipliers, $\alpha_i$, and it can be shown that the series coefficients in (2) can be written as $\beta_i = \alpha_i y_i$. The SVM model features a non-regularized bias term $b$ to be added to (1), and the solution is sparse in $\alpha_i$.

When dealing with the Regularized Least Squares model, the problem to be optimized is:

$$\min_f C\sum_{i=1}^{n_p}(y_i - f_i)^2 + \frac{1}{2}\|f\|_K^2 \quad (4)$$

where $C$ is a hyperparameter that plays the role of the coefficient $1/\lambda$, often adopted in the formalisms in literature [3]. The optimum can be found by solving a linear system. The RLS method does not involve a bias term, and the solution is not sparse due to the nature of the specific loss function.

In spite of those differences, both the SVM and the RLS approach adopt the same regularization term and share the basic idea of restricting the available space of functions. As a result, both regularization methods well fit the approach to the evaluation of Maximal discrepancy complexity, as it will be exposed in the following.

## III. Generalization Bounds

Two possible paths exist in order to quantify the generalization error for classifiers: a practical one, based on empirical methods such as Leave One Out or K-Fold Cross Validation, and a theoretical one, namely the computation of generalization bounds [6].

Theoretical approaches derive analytical expressions of the generalization bounds [6]. These methods do not require any data partitioning and are always based on a notion of Complexity on the hypothesis space, $\Lambda$. In view of the probabilistic nature of the involved quantities, a bound value is estimated with probability at least $1-\delta$, and is commonly written as:

$$\pi \le \nu + \chi + \tau \quad (5)$$

where $\pi$ is the 'true' generalization error, $\nu$ is the error on the training set, $\chi$ measures the complexity of the space of classifying functions, and $\tau$ penalizes the finiteness of the training sample. Generalization bounds differ in their dependency on the training set. Data-independent bounds from Statistical Learning Theory [1] relate the complexity term to a worst-case analysis. These bounds are often quite loose and the task of computing $\chi$ may prove quite difficult. For data-dependent bounds instead, the dependency from the training samples is embedded in the notion of Complexity that is adopted. The Maximal-Discrepancy bound [2][7] proves accurate and easy to be worked out, applying the following strategy:

I. the available training patterns are modified in such a way that a random half of the classes are swapped, whereas the remaining half remain unchanged [2][7].

II. a conventional RC is used to compute the average error rate, $\bar{v}$, on the modified training set.

For a decision function, $f$, drawn from the hypothesis space, $\Lambda$, the maximal discrepancy bound is defined as [8]:

$$\pi \le \nu + (1-2\bar{v}) + 3\sqrt{\frac{-\ln(\delta)}{2n_p}} \quad (6)$$

where the last term is the finite-sample penalty. The following discussion will only concentrate on the complexity term, i.e., the term $1-2\bar{v}$.

## IV. Maximal Discrepancy Precise Estimation

Complexity depends on $\bar{v}$, which can be computed by Montecarlo simulations as the average error of a learning machine trained with a half-swapped set of pattern classes. The learning machine is asked repeatedly to try to learn a common data set in which half of the classes are in turn assigned at random. In practice, the machine is asked to learn noise.

The meaning of (6) can be informally expressed as: "If the machine is able to fit noise, then the decision function is too complex". Hence, the machine will likely overfit data and the learning performance will degrade accordingly. This calls for some mechanism that can estimate and subsequently control complexity, in order to attain tighter bounds.

### A. *Effective complexity estimation*

The approach proposed in this paper relies on three key observations:

- When a learning machine is trying to understand noise, with the aim to get a low error, it will likely raise its internal complexity representation
- The causes of this complexity explosion can be identified in the nature of the kernel function adopted
- Information derived from training the learning machine on the original labels can be very useful in computing the bound.

When dealing with Regularized Classifiers, a regularized

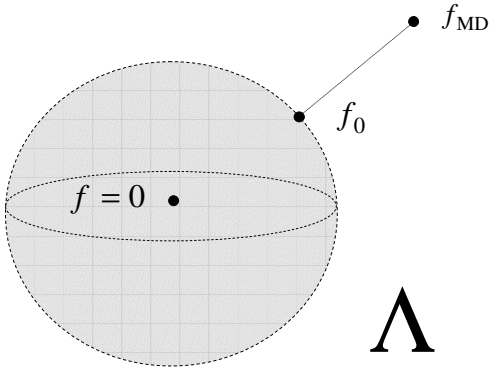


Fig. 1. Hypothesis space reduction: the high-complexity $f_{MD}$ model projects onto the hypersphere delimited by the original model $f_0$.

algorithm trying to learn noisy data will strongly increase its complexity, hence the regularization term $\| f \|_K^2$ can eventually increase uncontrollably. Empirical studies outlined that, for RC endowed with a linear kernel, the complexity explosion seldom takes place; by contrast, in the presence of an RBF kernel complexity tends to increase dramatically. This can be ascribed to the lower representation capabilities of a linear kernel with respect to the RBF one.

The basic idea underlying the present approach is to record the squared norm, $\| f_0 \|_K^2$ , obtained as a result of training an RC on the original data. This norm represents the level of complexity reached by the RC when dealing with data and original labels. In other words this means that this is the level of complexity that requires this problem. Such a level can be considered as a bound for the RC when involved in the bound computation.

Formally one notes that the following optimization problem is equivalent to the original RC problem:

$$\min_f C\, R_{emp}[f] + \frac{1}{2}\| f \|_K^2 \quad \text{s.t. } \| f \|_K^2 \le \| f_0 \|_K^2 \tag{7}$$

When real labels are used, the additional inequality is always satisfied, and there is no risk that (7) yields a solution different from $f_0$. As long as one is certain that the bound on $f_0$ is satisfied (Fig.1), the approach (7) can be used, in place of the original, unconstrained RC formulation, to accomplish the Montecarlo iterative run that produces $\overline{\nu}$ .

The additional constraint has a major implication: it prevents the explosion in complexity whenever the RC involved in step II tries to lower the training error on the randomly labeled data. As a consequence of being more constrained, each RC in the Montecarlo series exhibits an equal or possibly worse error rate than its unconstrained counterpart. Since the training error $\overline{\nu}$ does not decrease, the term $(1-2\overline{\nu})$ becomes smaller or at least unchanged, hence the bound (6) shrinks accordingly.

Summarizing, a complexity reduction strategy is not applied to the learning machine when coping with the original problem but to the machine involved in the bound computation. This, in other words, means limiting the complexity of the trained RC during the bound computation.

Important features of the overall approach are that it does not imply any particular hypothesis, and that it leads to a precise estimation of a Maximal-discrepancy measure of complexity.

### *B. A practical Algorithm*

The proposed formulation (7) offers the additional advantage of a straightforward method to perform the training process, allowing the use of established and effective existing tools. The algorithm progressively decreases the hyperparameter *C* (e.g. by 10% of the current value of C, see $\vartheta_C$ in the pseudo-code) until the constraint $\| f \|_K^2 \le \| f_0 \|_K^2$ is satisfied.

*Precise Maximal Discrepancy Complexity Estimation*

Inputs: $n_p$ input patterns $(X_i, y_i)$, where $y_i$ is the target value and i = 1,… ,np,

Parameters: a regularization constant C, the RBF kernel width $\sigma$

Output: the complexity term $1-2\overline{\nu}$

1. Train the machine for the original problem:
   - 1.1. Solve: $\boldsymbol{\beta}$ = RCTraining$(C, 2\sigma^2, \mathbf{y}, \mathbf{X})$
   - 1.2. Compute and record: $\| f_0 \|_K^2 = \boldsymbol{\beta}^t \mathbf{K} \boldsymbol{\beta}$ ;
   - 1.3. Set $\kappa$ := 0
2. For $i$ = 1 : $\eta$
   - 2.1. Random swap half of the pattern labels
     - 2.1.1. Set $\hat{C} := C$ ;
     - 2.1.2. $\hat{\mathbf{y}}$ = getHalfRndSwapLabels()
   - 2.2. Train a RC on the new problem:
     - 2.2.1. Solve: $\hat{\boldsymbol{\beta}}$ =RCTraining$(\hat{C}, 2\sigma^2, \hat{\mathbf{y}}, \mathbf{X})$
     - 2.2.2. Compute: $\| f_{MD} \|_K^2 = \hat{\boldsymbol{\beta}}^t \mathbf{K} \hat{\boldsymbol{\beta}}$
     - 2.2.3. Compute $\nu$ (training error rate)
   - 2.3. **If** $(\| f_{MD} \|_K^2 \le \| f_0 \|_K^2)$ **goto** 2.4 **else** Set: $\hat{C} = \hat{C} / \vartheta_C$ ; **goto** 2.2
   - 2.4. Compute: $\kappa = \kappa + \nu$
3. $\overline{\nu} = \kappa / \eta$
4. **return** $1-2\overline{\nu}$

Such an approach allows one to apply any existing, efficient algorithm (in the following called `RCTraining`) to the training step of the RCs, whose learning phase is the central core of the iterative procedure. In the presented pseudocode, $\mathbf{X}$ is the data matrix, $\mathbf{K}$ is the kernel matrix, $\boldsymbol{\beta}$ and $\hat{\boldsymbol{\beta}}$ are the expansion coefficients in (2) of the trained RC with the real $\mathbf{y}$ and the swapped labels $\hat{\mathbf{y}}$, respectively, $2\sigma^2$ is the Gaussian kernel parameter and $\eta$ is the number of runs. Finally, $\| f_{\mathrm{MD}} \|_K^2$ represents the norm of the running function learned in the Discrepancy Bound computation, which complexity is iteratively bounded by the reference $\| f_0 \|_K^2$.

## V. Experimental results

The effectiveness and the generality of the proposed approach were tested for both of the cited models of regularized classifiers, namely, the Support Vector Machine and the Regularized Least Squares method. Both classifiers were trained on several dataset to investigate the ability of the methodology to cope with different problems.

The experiments reported here involved three UCI datasets [8]: Pima-Indians Diabetes (768 patterns), Ionosphere (351 patterns), and Sonar (208 patterns). All these datasets present binary classification problems.

For both classifiers, the three experiments were performed under the same experimental conditions:

1) All data were normalized between [-1,+1];
2) The RBF kernel was adopted:
$$K(\mathbf{x_1}, \mathbf{x_2}) = e^{-\|\mathbf{x_1}-\mathbf{x_2}\|^2 / 2\sigma^2}$$
3) Model selection was performed by training the regularized machines for every combination of values of both hyperparameters:
   - $C = \{0.1, 1, 100\}$
   - $2\sigma^2 = \{0.1, 1, 10, 100, 1000\}$.
4) Each computation of the value $\bar{v}$ resulted from the average of over 100 runs ($\eta = 100$),
5) The decrease rate of $C$ (as per step 2.3) $\vartheta_C$ was 1.1

Results are reported separately for the two tested classifiers. In order to provide an overview of the full process, the following presentation gives the complete set of Maximal discrepancy bounds computed with both the classical method and the proposed methodology.

### A. Effectiveness evaluation for SVM

The SMO algorithm was used as the Support Vector Machine optimizer. The tolerance on KKT conditions was $10^{-3}$. The developed SMO algorithm uses a first order heuristic for working set selection [5]. Figure 2 reports on the computed complexity terms for the Diabetes (2.a), Sonar (2.b) and Ionosphere (2.c) datasets, respectively. The figures show that the proposed approach allows computing theoretical bounds at least as precise as those obtained by using the classical approach. Table I gives a summary of the obtained results for the SVM classifiers, and outlines that the average improvement in bound precision is remarkable.

TABLE I
Performance Evaluation for the Proposed Method (SVM)

| Dataset | np | Avg bound SVM | Avg bound SVM$f_0$ | Maximum advantage | Average advantage |
|---|---|---|---|---|---|
| Diabete | 768 | 23.02% | 21.80% | 6.50% | 1.22% |
| Sonar | 208 | 49.18% | 47.08% | 22.50% | 2.10% |
| Ionosphere | 351 | 39.59% | 35.44% | 26.97% | 4.15% |

### B. Effectiveness evaluation for RLS

The Regularized Least Squares was implemented by solving the system of equations induced by problem (4) by using standard Matlab implementation of the Gaussian Elimination algorithm. Figure 3 reports the theoretical complexity computed by using both the classical and the proposed method for the three considered datasets.

As for the previous classifier, the proposed method allows computing tighter bounds. Even though the general improvement (Table II) is less consistent than that obtained for the SVM, the methodology presented in this paper proves to be effective also when applied to the regularized least squares classifier.

TABLE II
Performance Evaluation for the Proposed Method (RLS)

| Dataset | np | Avg bound RLS | Avg bound RLS$f_0$ | Maximum advantage | Average advantage |
|---|---|---|---|---|---|
| Diabete | 768 | 33.76% | 33.20% | 4.88% | 0.56% |
| Sonar | 208 | 69.46% | 68.82% | 6.36% | 0.64% |
| Ionosphere | 351 | 57.47% | 56.06% | 12.30% | 1.42% |

## VI. Conclusions

The paper introduces a simple method that leads to a precise Maximal discrepancy estimation of complexity. Theory and experiments underline the ability of the method to work with two well-known machine models such as SVM and RLS. The proposed method is designed to be effective when using RBF kernels. Therefore, it is suitable to handle also highly non-linear mappings.

The obtained results prove that, when complexity tends to increase significantly, a consistent compensating reduction effect can be obtained. On average, the presented experiments seem to point out that the SVM model benefits from the reduction induced by the constrained approach more than the RLS model.

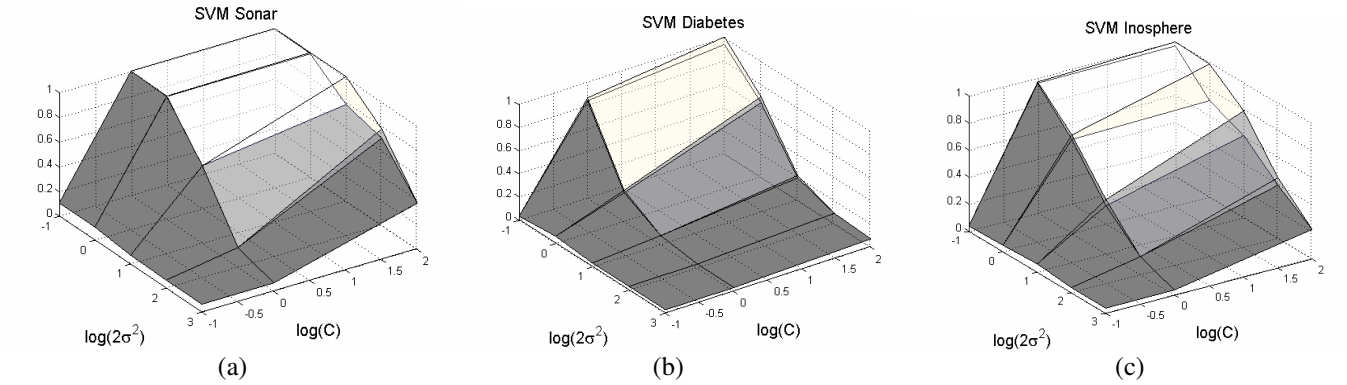


Fig. 2. Complexity estimation for SVM with classical and precise estimation. The transparence reveals the underlying surfaces obtained with the precise estimation method. (a) Sonar, (b) Diabetes, (c) Ionosphere

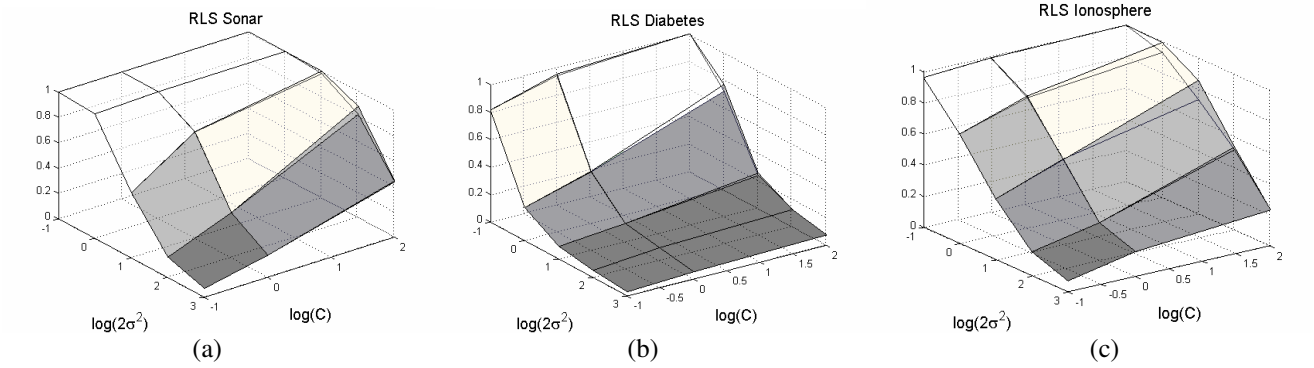


Fig. 3. Complexity estimation for RLS with classical and precise estimation. The transparence reveals the underlying surfaces obtained with the precise estimation method. (a) Sonar, (b) Diabetes, (c) Ionosphere

Although the gain in tightening bounds is not impressive, from the theoretical point of view this work underlines a salient feature: maximal discrepancy bound evaluation is strongly affected by the complexity of the single classifiers trained during the Montecarlo computations. Future work can address this aspect more in dept.

## REFERENCES


[1] Vapnik V. "Statistical Learning Theory", 1998, Wiley-Interscience Pub.

[2] P. L. Bartlett, S. Boucheron, G. Lugosi "Model Selection and Error Estimation" , Machine Learning 48(1-3), Springer, 2002

[3] Schölkopf, B. and A.J. Smola: "Learning with Kernels". MIT Press, Cambridge, MA, USA (2002)

[4] T. Evgeniou, M. Pontil, T. Poggio "Regularization Networks and Support Vector Machines" Advances in Computational Mathematics 13(1), Springer, 2000.

[5] C.C.Chang, C.J. Lin . "LibSVM: a library for Support Vector Machines" [http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf]

[6] D. Anguita, S. Ridella, F. Rivieccio, R. Zunino "Hyperparameter tuning criteria for support vector classifiers", Neurocomputing, October 2003, 55, pp. 109-134.

[7] D. Anguita, A. Boni, S. Ridella, F. Rivieccio, D. Sterpi "Theoretical and Practical Model Selection Methods for Support Vector Classifiers" in L. Wang (Ed.), Support Vector Machines: Theory and Applications

[8] Hettich, S. and Bay, S. D. (1999). The UCI KDD Archive [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science