Using Unsupervised Analysis to Constrain Generalization Bounds for Support Vector Classifiers

Sergio Decherchi, *Student Member, IEEE*, Sandro Ridella, *Member, IEEE*, Rodolfo Zunino, *Senior Member IEEE*, Paolo Gastaldo, and Davide Anguita, *Member, IEEE*.

***Abstract*— A crucial issue in designing learning machines is to select the correct model parameters. When the number of available samples is small, theoretical sample-based generalization bounds can prove effective, provided that they are tight and track the validation error correctly. The Maximal Discrepancy approach is a very promising technique for model selection for Support Vector Machines (SVM), and estimates a classifier's generalization performance by multiple training cycles on random labeled data. This paper presents a general method to compute the generalization bounds for SVMs, which is based on referring the SVM parameters to an unsupervised solution, and shows that such an approach yields tight bounds and attains effective model selection. When one estimates the generalization error, one uses an unsupervised reference to constrain the complexity of the learning machine, thereby possibly decreasing sharply the number of admissible hypothesis. Although the methodology has a general value, the method described in the paper adopts Vector Quantization (VQ) as a representation paradigm, and introduces a biased regularization approach in bound computation and learning. Experimental results validate the proposed method on complex real-world data sets.**

***Index Terms*— Support Vector Machine, Vector Quantization, Maximal Discrepancy, Mixed Unsupervised and Supervised methods**

I. Introduction

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, $\widetilde{\pi}$, results from the sum of two terms: the empirical error on the training sample, ν, and a penalty term, $\Delta\pi$, associated with the model complexity; the underlying epistemic assumptions is that complex models are exposed to the risk of overfitting data, hence should be subject to high generalization bounds:

$$\widetilde{\pi} = \nu + \Delta\pi \tag{1}$$

The first term in the right-hand side of (1) depends on training data, whereas the second term should not depend on the classes of training patterns. In the case of Support Vector Machines for classification [1,2], the literature offers a variety of theoretical approaches to attain an (usually upper-bounded) estimate of the generalization error [3,4]. This proves especially useful in the presence of limited training samples, when classical techniques such as Cross-Validation [5] may waste samples because one has to exclude training patterns from the parameter adjustment. The Maximal-Discrepancy (MD) approach [6] is often effective in those cases for estimating the generalization penalty, $\Delta\pi$. 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, $\Delta\pi$, of the bounds themselves.

Within that framework, this paper introduces a novel model (VQSVM) to support the computation of tighter penalty terms, $\Delta\pi$. 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, $\Delta\pi$, for the basic classifier, any contribution is generated by a set of possible hypothesis that are constrained by the 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 [7,8,9]; in the present research (and also in [10,11]) 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. The VQSVM methodology, however, features a wider general validity and is not tied to either of those approaches. The theoretical analysis shows that the method always improves on, or at least is equivalent to the classical Structural Risk Minimization paradigm [1], whose results reduce to a special (worst-) case of the VQSVM

Manuscript received January 30, 2009; Revised June 29, 2009; Accepted November 28, 2009.

Authors are with the Department of Biophysical and Electronics Engineering (DIBE), at Genoa University, Italy. Email: sergio.decherchi@unige.it phone +39 0103532269.

predictions.

At the same time, the paper proves the convexity of the VQ-constrained problem formulation; the (global) optimal solution is always achieved when one minimizes the VQSVM functional to compute the generalization bound. Finally, the theoretical analysis describes an efficient procedure to compute the penalty term under the unsupervised-constrained assumption.

The method effectiveness is demonstrated on different, real-world testbeds: the NIST dataset [12], the Newsgroup-20 text-mining dataset [13], and several other UCI datasets [13]. 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.

The paper is organized as follows: Section II briefly overviews the overall theoretical background; Section III illustrates the actual use of unsupervised clustering for classifier design and Section IV describes the optimization problem. Experimental results are presented in Section V. Some concluding remarks are made in Section VI.

II. Theoretical Background

A. Notation for Support Vector Machines in classification problems

A binary classification problem involves using a set of labeled patterns $Z = \{(\mathbf{x}_i, y_i);\ i = 1,..,np;\ y_i \in \{-1, +1\}\}$. The Support Vector Machine model [2] proves a valuable algorithm for that task [14], and requires the solution of a Quadratic Programming problem:

$$\min_{\alpha}\left\{\frac{1}{2}\sum_{i,j=1}^{np}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j) - \sum_{i=1}^{np}\alpha_i\right\}$$

$$\text{subject to: } \begin{cases} 0 \le \alpha_i \le C, \forall i \\ \sum_{i=1}^{np} y_i\alpha_i = 0 \end{cases} \tag{2}$$

where $\{\alpha_i\}$ are the SVM parameters setting the class separating surface, the scalar quantity C upper bounds the SVM parameters, and $K(\cdot,\cdot)$ is the kernel function, i.e., a basis for the SVM series expansion. If $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$ are the points in the "feature" space associated with $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively, then their dot product can be written as $\langle\Phi(\mathbf{x}_i),\Phi(\mathbf{x}_j)\rangle = K(\mathbf{x}_i,\mathbf{x}_j)$. An SVM supports a linear class separation in that feature space; the classification rule for a trained SVM is:

$$f(\mathbf{x}) = \sum_{i=1}^{np}\alpha_i y_i K(\mathbf{x},\mathbf{x}_i) + b \tag{3}$$

where b is a bias term and can be computed by using the set of values $\{\alpha_i\}$ [2]. The set of patterns, including $n_{SV} \le np$ "support vectors", such that the associate parameters α_i are non-null can be found by solving (2). In the Hilbert space, these vectors span a hyperplane that separates classes and is expressed as:

$$\mathbf{w}^{(\mathrm{TG})} = \sum_{i=1}^{np} y_i\alpha_i\Phi(\mathbf{x}_i) \tag{4}$$

For an exhaustive presentation of the method, see [1]. The complexity of the hypersurface (3) is affected by both the value of C and the specific kernel adopted.

B. Maximum Discrepancy Generalization Bounds

Formal approaches to predicting a classifier's generalization error [1] often prove impractical, mainly due to the loose bounds obtained. Conversely, empirical methods such as cross-validation and k-fold cross validation [15] can attain useful estimates at the cost of reducing the number of training patterns.

Maximum-Discrepancy [6] methods for assessing generalization bounds aim to ensure the validity of analytical predictions while retaining the advantages of empirical estimates. In a MD procedure, one splits the available dataset, Z, into two halves (suppose np is even), and computes the fractions $\{\nu_1, \nu_2\}$ of misclassified patterns for the two halves:

$$\begin{aligned} \nu_1 &= \tfrac{2}{np}\sum_{i=1}^{np/2} L[f(\mathbf{x}_i), y_i] \\ \nu_2 &= \tfrac{2}{np}\sum_{i=np/2+1}^{np} L[f(\mathbf{x}_i), y_i] \end{aligned} \tag{5}$$

where $L[\cdot,\cdot]$ is the classical 0-1 loss function, and $f(\mathbf{x})$ is a model belonging to the *hypothesis space* Λ, (i.e., the space of admissible models resulting from a learning process). Finally, one computes a random variable, called Maximal Discrepancy, as:

$$\xi_Z = \max_{\Lambda}(\nu_2 - \nu_1) \tag{6}$$

The definition of ξ_Z leads to a tight, data-dependent generalization bound: if one denotes with π and ν the true generalization error and the classification error on the training set, respectively, theory shows [6] that, with probability higher than 1-δ, the following bound holds true:

$$\pi \le \nu + \xi_Z + \Delta_B(np,\delta) \tag{7}$$

where the Bernoulli correction term, Δ_B, in (7) only depends on the training set cardinality: $\Delta_B(np,\delta) = 3\sqrt{\ln(\delta^{-1})/(2np)}$. The method presented in [6] gives a straightforward way to compute ξ_Z: one flips the classes in one of the two halves, leaving the other half unchanged; then one trains a conventional classifier on this modified dataset and measures

the empirical training error, $\overline{\nu}$. If $\overline{\nu}_2$ denotes the classification error on the fraction of patterns with flipped targets, then clearly one has: $\nu_2 = 1 - \overline{\nu}_2$; therefore the penalty term (6) can be written as:

$$\xi_Z = \max_{\Lambda}(\nu_2 - \nu_1) = \max_{\Lambda}(1 - \overline{\nu}_2 - \nu_1) = \\ = 1 - \min_{\Lambda}(\overline{\nu}_2 + \nu_1) = 1 - 2\overline{\nu} \tag{8}$$

Thus the penalty term (6) is worked out by minimizing the classification error, $\overline{\nu}$, on the modified training set with a conventional SVM training process. The computation (8) is iterated several times by random flipping (a half of) the pattern classes to attain a robust estimation of the penalty term (6). As compared with data-independent bounds resulting from worst-case analysis [1], the data-dependent bound (7) usually proves tighter [6]. From a strictly theoretical viewpoint, Maximum Discrepancy [6] does not completely adhere to the SVM model, as the SVM loss function is not 0-1; nevertheless, it turns to be very effective and reliable [16]. For these reasons, the approach presented in this paper relies on ξ_Z as an estimator to drive the complexity-reduction process.

III. Constraining SVM Capacity By Unsupervised Analysis

The VQSVM method proposed in this paper is effective in the computation process of the penalty term (6), 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, ν, in the generalization estimate (7), which always depends on training data 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, $\Delta\pi$; 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.

A. 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 [17-19]; 2) a Support Vector-based representation (3) of the clustering results.

Step 1) The Kernel *k*-means algorithm [17-19] divides the projections, Φ_i, of input data into two clusters, C_u (u =0,1). The algorithm operates on distance values that are computed by using the kernel trick without the explicit coordinates of cluster centroids, Ψ_u. Let the "membership vector" $\mathbf{m} \in \{0,1\}^{np}$ encode the partitioning of input patterns into the two clusters: $m_i = 0$ if $\Phi_i \in C_0$ and $m_i = 1$ if $\Phi_i \in C_1$, $i = 1,\ldots, np$. Each prototype lies in the centroid of its associate partition, hence the membership vector **m** determines the prototypes' positions even though they are not stated explicitly. For a pattern $\mathbf{x}_i$, the square distance, d, from its image, Φ_i, to Ψ_u is worked out as:

$$d(\Psi_u, \Phi_i) \\ = \left(\frac{1}{np_u} \sum_{j=1}^{np} \Phi_j \delta_{ju} \right)^2 + {\Phi_i}^2 - \frac{2}{np_u} \sum_{j=1}^{np} \delta_{ju} (\Phi_i \cdot \Phi_j) = \\ = \frac{1}{np_u^2} \sum_{j,k=1}^{np} \delta_{ju}\delta_{ku} K(\mathbf{x}_j, \mathbf{x}_k) + K(\mathbf{x}_i, \mathbf{x}_i) - \frac{2}{np_u} \sum_{j=1}^{np} \delta_{ju} K(\mathbf{x}_i, \mathbf{x}_j) \tag{9}$$

where u =0, 1, δ_{ju}=1 if m_j=u, and 0 otherwise; $np_u = \sum_{j=1}^{np} \delta_{ju}$ is the number of patterns in the cluster C_u. By using expression (9), which includes only kernel computations, one identifies the closest prototype to the image of each input pattern, and ascribes the pattern membership accordingly:

The feature-space version of k-means clustering

---

1. *Initialize* **m** with random memberships $m_i \in \{0,1\}$; mark **m** as 'modified'
2. *While* (**m** is modified):
   a. Compute distances: $d(\Psi_u, \Phi_i)$, $u = 0,1$; $i=1,\ldots,np$;
   b. Update **m** such that: $m_i = \arg\min_u d(\Psi_u, \Phi_i)$

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: $Z^+ = \left\{ \left(\mathbf{x}_i, y_i^{(\mathrm{KM})+} \right),\ i = 1,\ldots,np;\ y_i^{(\mathrm{KM})+} = 2m_i - 1 \right\}$. This set undergoes a conventional SVM training process (2). The resulting hyper-plane, $\mathbf{w}^{(\mathrm{KM})+}$, is given by:

$$\mathbf{w}^{(\mathrm{KM})+} = \sum_{i=1}^{np} y_i^{(\mathrm{KM})+} \alpha_i^{(\mathrm{KM})+} \Phi(\mathbf{x}_i) \qquad (10)$$

where the coefficients $\{\alpha_i^{(\mathrm{KM})+}\}$ are associated with cluster-related provisional classes, $y_i^{(\mathrm{KM})+}$. Then one builds up the dual training set, which supports the opposite labeling schema: $Z^- = \{(\mathbf{x}_i, y_i^{(\mathrm{KM})-}),\ i = 1,\ldots,np;\ y_i^{(\mathrm{KM})-} = 1 - 2m_i\}$, and obtains the alternative parameters, $\mathbf{w}^{(\mathrm{KM})-}$, as:

$$\mathbf{w}^{(\mathrm{KM})-} = -\mathbf{w}^{(\mathrm{KM})+} \qquad (11)$$

To choose between the two alternatives $\{\mathbf{w}^{(\mathrm{KM})+}, \mathbf{w}^{(\mathrm{KM})-}\}$, one follows the Structural Risk Minimization principle, and picks the labeling schema that better constrains complexity. If one denotes with $\mathbf{w}^{(\mathrm{TG})}$ the solution obtained by using real labels, and with $\mathbf{w}^{(\mathrm{KM})}$ the unsupervised "reference" solution, the latter parameter set that further constrains SVM capacity is:

$$\mathbf{w}^{(\mathrm{KM})} = \arg\min\left(\left\| \mathbf{w}^{(\mathrm{KM})+} - \mathbf{w}^{(\mathrm{TG})} \right\|^2, \left\| \mathbf{w}^{(\mathrm{KM})-} - \mathbf{w}^{(\mathrm{TG})} \right\| \right) \qquad (12)$$

Upon completion of the unsupervised analysis, it is convenient to use the reference solution, $\mathbf{w}^{(\mathrm{KM})}$, and the artificial target settings, $y^{(\mathrm{KM})}$, adopted to attain the unsupervised SVM training, to compute the following quantities:

$$\beta_i = y_i \sum_j \alpha_j^{(\mathrm{KM})} y_j^{(\mathrm{KM})} k(\mathbf{x_i}, \mathbf{x_j});\quad i = 1, \ldots, np \qquad (13)$$

This set of parameters will be used later in the theoretical treatment.

B. Using unsupervised clustering to constrain SVM capacity during bound computation

The result (12) of the unsupervised analysis is used in the computation of the MD bounding term (8) on random targets, and sets a reference to arrange the family of classifiers within the hypothesis space, Λ. The quantity ruling the ordering of classifiers is the distance, in the weight space, between a given SVM solution, **w**, and the reference configuration, $\mathbf{w}^{(\mathrm{KM})}$:

$$\rho_{\mathbf{w}} = \left\| \mathbf{w} - \mathbf{w}^{(\mathrm{KM})} \right\| \qquad (14)$$

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, $\mathbf{w}^{(\mathrm{KM})}$, and the real classification surface, $\mathbf{w}^{(\mathrm{TG})}$, must coincide. Of course, the opposite case may occur, in which the target distribution is totally uncorrelated with the obtained clusters; those extreme situations are illustrated in Fig.1. The varying displacements

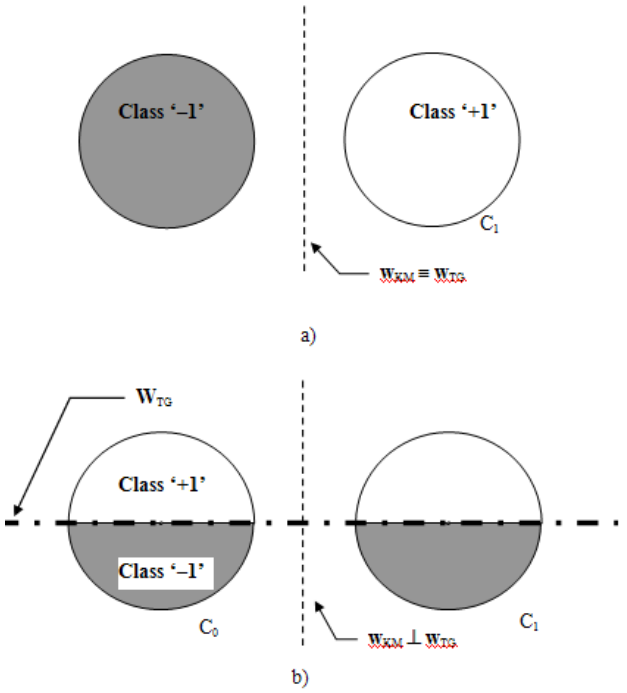


Fig.1 Opposite situations may arise from unsupervised analysis: a) Consistent case: clusters match class distribution. b) Opposite case: clusters do not match classes.

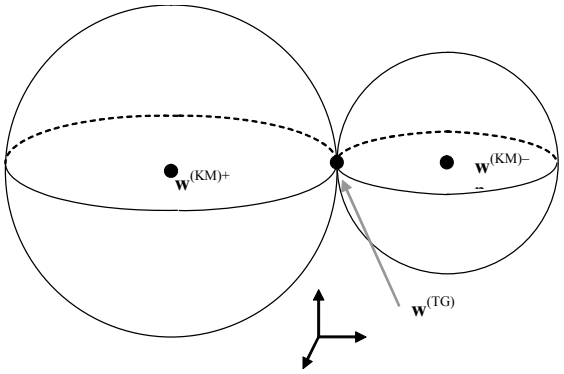


Fig.2 The original target solution, $\mathbf{w}^{(\mathrm{TG})}$, and the candidate centers for unsupervised reference, $\mathbf{w}^{(\mathrm{KM})\pm}$. The spheres intersect at $\mathbf{w}^{(\mathrm{TG})}$ and the smaller radius determines the eventual reference, $\mathbf{w}^{(\mathrm{KM})}$.

obtained from empirical results can give useful information about the complexity of the specific classification problem.

Such a distance-based ordering is profitable in the Maximal-Discrepancy approach to limit the number of admissible solutions. For a given value of $\rho_{\mathbf{w}}$, only the classifier configurations lying within the hypersphere having radius $\rho_{\mathbf{w}}$ and centered in $\mathbf{w}^{(\mathrm{KM})}$, will be considered to compute the penalty term (8). Larger and larger spheres enable the training algorithm to pick the optimal weight set, **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, $\rho_{\mathbf{w}}$, of the sphere that includes the admissible solutions of (8) is the crucial quantity driving the SRM principle (Fig.2), and the proper setting of such a regularization parameter is of paramount importance. The VQSVM approach uses the SVM solution, $\mathbf{w}^{(\mathrm{TG})}$, obtained on the real labels to delimit the valid portion of the weight space for the admissible solutions of (8). The regularization parameter is set as:

$$\rho_0 \overset{def}{=} \left\| \mathbf{w}^{(\mathrm{TG})} - \mathbf{w}^{(\mathrm{KM})} \right\| \tag{15}$$

In other words, every solution, **w**, obtained during the MD estimation process must obey the distance-based criterion:

$$\rho_{\mathbf{w}} \le \rho_0 \tag{16}$$

and the optimization problem to be solved is expressed by (2) under the additional constraint (16). This poses two major questions.

The first question regards the effectiveness of constraint (16) in bounding the generalization error. In the most favorable case, one has $\mathbf{w}^{(\mathrm{TG})} \equiv \mathbf{w}^{(\mathrm{KM})}$, as per Fig.1a): the separating surface drawn by unsupervised clustering on artificial targets coincides with that obtained with real targets, hence one has: $\max(\rho_{\mathbf{w}}) = 0$. When the empirical classifier matches the natural distribution of data, the number of allowed family members reduces to one, and the associate penalty term, ξ_Z, theoretically vanishes. By contrast, the worst situation occurs when $\left\| \mathbf{w}^{(\mathrm{TG})} - \mathbf{w}^{(\mathrm{KM})} \right\| \approx \infty$. As the hypersphere encompasses the entire weight space, all classifiers in Λ 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 optimization process to solve the reformulated problem (2)+(16). The crucial issue is that (16) 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 (2)+(16), which can still take advantage of a Quadratic-Programming formulation and ensure convergence to the global, optimal solution.

C. Including the additional constraint into the SVM optimization problem

The modified Primal formulation includes the quadratic constraint (16) in the SVM basic problem setting to solve (2) on the random targets assigned by the MD procedure, and can be written as:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} P_M = {} & \frac{\|\mathbf{w}\|^2}{2} + C\sum_{i=1}^{np}\xi_i + \\ & - \sum_{i=1}^{np}\alpha_i\{y_i(\mathbf{w}\mathbf{x}_i + b) - 1 + \xi_i\} + \\ & - \sum_{i=1}^{np}\gamma_i\xi_i - \frac{\lambda}{2}\left[\rho_0^2 - \left\|\mathbf{w} - \mathbf{w}^{(\mathrm{KM})}\right\|^2\right] \end{aligned} \tag{17}$$

where P_M stands for Modified Primal, and λ is the Lagrange multiplier associated to the constraint (16). One derives a Dual problem formulation [1] by nullifying the partial derivatives with respect to the optimization variables:

$$\frac{\partial P_M}{\partial w_i} = w_i - \sum_j \alpha_j y_j \Phi(x_j)_i + \lambda\left(w_i - w_i^{(\mathrm{KM})}\right) = 0 \tag{18a}$$

$$\frac{\partial P_M}{\partial b} = \sum_j \alpha_j y_j = 0 \tag{18b}$$

$$\frac{\partial P_M}{\partial \xi_i} = C - \alpha_i - \gamma_i = 0 \tag{18c}$$

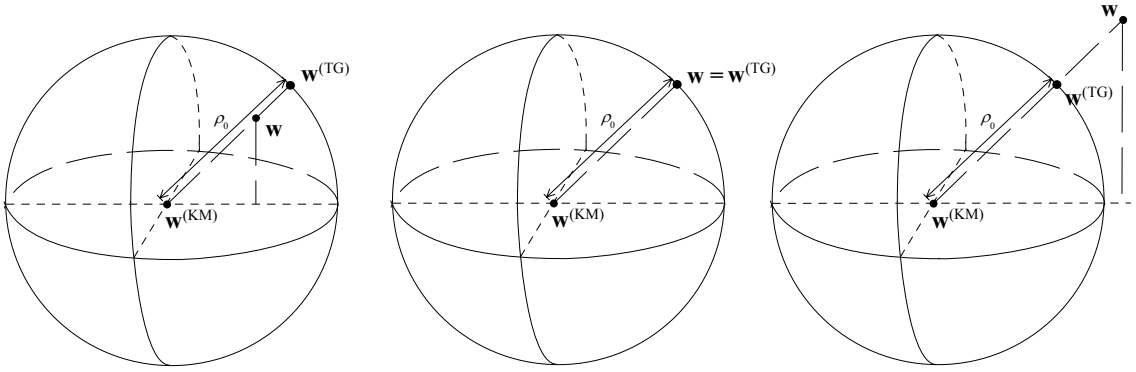
By solving (18a) with respect to the weight components, w_i, one obtains:

$$\mathbf{w} = \frac{\sum_i \alpha_i y_i \Phi(\mathbf{x}_i) + \lambda \mathbf{w}^{(\mathrm{KM})}}{(1+\lambda)} \tag{19}$$

When the multiplier λ is zero, constraint (16) is inactive and the solution, **w**, takes back the form of the basic SVM parameters; this means that the solution lies inside the sphere (16). In the following, the symbol $\mathbf{w}^{(\lambda 0)}$ will denote the weight vector associated with this case:

$$\mathbf{w}^{(\lambda 0)} = \sum_{i=1}^{np} \alpha_i y_i \Phi(\mathbf{x}_i) \tag{20}$$

Expression (20) holds for any class configuration $\{y_i\}$, and embeds the part of the solution which does not consider the quadratic constraint (16). Rewriting problem $\mathrm{P_M}$ into its Dual formulation leads to the optimization problem:



a) b) c)

Fig.3 Relative positions of the solution vector, **w**, with respect to the unsupervised reference, $\mathbf{w}^{(KM)}$ a) Case 1): Within the hypersphere; b) Case 2: on the hypersurface; c) Case 3: Out of the hypersphere

$$\min_{\alpha,\lambda} D_M = \left(\frac{\left\| \mathbf{w}^{(\lambda 0)} \right\|^2}{2} - \sum_i \alpha_i \right) + \frac{\lambda}{2} \left[\rho_0^2 - \frac{\left\| \mathbf{w}^{(\lambda 0)} - \mathbf{w}^{(\mathrm{KM})} \right\|^2}{(1+\lambda)} \right] \overset{def}{=} \overset{def}{=} D_{\mathrm{M,SVM}} + D_{\mathrm{M},\lambda}$$
$$\text{subject to}: \begin{cases} 0 \le \alpha_i \le C \quad \forall i = 1,\ldots,np \\ \sum_i \alpha_i y_i = 0 \quad \forall i = 1,\ldots,np \\ \lambda \ge 0 \end{cases} \tag{21}$$

The cost function in (21) comprises two terms: the left term, denoted as $D_{\mathrm{M,SVM}}$, 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 $D_{\mathrm{M},\lambda}$, is parameterized by λ and takes into account the contribution of the quadratic constraint:

$$\min_{\alpha,\lambda} D_M = \min_{\alpha,\lambda} [D_{\mathrm{M,SVM}}(\alpha) + D_{\mathrm{M},\lambda}(\alpha,\lambda)] \tag{22}$$

After simple derivations and substitutions the dual optimization problem is eventually written as:

$$\min_{\alpha,\lambda} \left\{ \left(\frac{\left\| \mathbf{w}^{(\lambda 0)} \right\|^2}{2} - \sum_i \alpha_i \right) + \frac{\lambda}{2} \left[\rho_0^2 - \frac{\left\| \mathbf{w}^{(\lambda 0)} - \mathbf{w}^{(\mathrm{KM})} \right\|^2}{(1+\lambda)} \right] + - \sum_i \mu_i \alpha_i - \sum_i \xi_i (C - \alpha_i) + b \sum_i \alpha_i y_i - \delta\lambda \right\} \tag{23}$$

The additional variables in (23) embed the constraints (16); 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 (2) without the additional constraint, then check if condition (16) is fulfilled; this may result in three different cases, depending on the position of the solution vector, **w**.

Case 1) (Fig.3a) The solution lies inside the sphere (16), hence the constraint is inactive and λ=0. The Modified problem (21) reduces to the conventional form, and the vector **w** is a valid solution.

Case 2) (Fig.3b) 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 **w** is a valid solution.

Case 3) (Fig.3c) The solution is out of the sphere, λ>0, the constraint is active, and one has:

$$\left\| \mathbf{w} - \mathbf{w}^{(\mathrm{KM})} \right\|^2 > \rho_0^2 \tag{24}$$

In this case, to reach a valid solution one requires an ad-hoc optimization process, which will be presented in the following Section.

IV. Algorithm for Constrained Optimization

The optimization of the training problem involved by eq.(21) 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 (21) and relies on any off-the-shelf SVM optimizer (e.g. SMO [22]), thus yielding a straightforward algorithmic implementation.

A. 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 (21) is convex. The following Theorem confirms this fact by proving that the associate Hessian matrix, **H**, is positive semi-definite. It is convenient in the following to use a compact notation and define the matrix **Q,** having size $np \times np$, as the matrix composed by the elements:

$$q_{ij} = y_i y_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) \tag{25}$$

Theorem 1: The Modified Dual functional (21) 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 (21) 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 (e.g., SMO [20]). The algorithm proceeds by alternating two steps. The first step minimizes $D_{\mathrm{M},\lambda}$ and works out the optimal value, $\tilde{\lambda}$, of the Lagrange multiplier when the parameters, $\boldsymbol{\alpha}$, remain fixed in (21). The following Lemma gives the analytical expression of $\tilde{\lambda}$, in which the apex T denotes the vector-transpose operation.

Lemma 1 : The optimal value, $\tilde{\lambda}$, that minimizes $D_{\mathrm{M},\lambda}$ for a fixed set of parameters, $\boldsymbol{\alpha}$, is:

$$\tilde{\lambda} = \sqrt{\frac{\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha} + \| \mathbf{w}_{KM} \|^2 - 2\left[\mathbf{w}^{(\lambda 0)}\right]^{\mathrm{T}} \mathbf{w}^{(\mathrm{KM})}}{\rho_0^2}} - 1 \tag{26}$$

Proof: the proof is given in the Appendix.

The second step minimizes the functional (21) over the parameters, $\boldsymbol{\alpha}$, 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 2 : For a fixed value $\tilde{\lambda}$, the quadratic convex cost D_M to be minimized can be written as:

$$\begin{aligned} &\frac{1}{2}\boldsymbol{\alpha}'\mathbf{Q}\boldsymbol{\alpha} - \sum_i \alpha_i (1 + \tilde{\lambda}(1 - \beta_i)) \\ &0 \le \alpha_i \le C \ \ \forall i = 1,...,np \\ &\sum_i \alpha_i y_i = 0 \ \ \forall i = 1,...,np \end{aligned} \tag{27}$$

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 (21); this multiplier is worked out as a side result of the minimization process of (21).

The procedure alternating Lemma 1 and Lemma 2 iterates until a solution lying within the hypersphere (16) is found. The following Theorem proves that such a procedure always converges to the global minimum.

Theorem 2 – A procedure alternating the partial optimizations as per Lemma 1 and Lemma 2, always reaches the global minimum of the Modified Dual cost eq.(21).

Proof. Lemma 1 and Lemma 2 ensure the minimization of $D_{\mathrm{M,SVM}}$ and $D_{\mathrm{M},\lambda}$ when the multiplier $\tilde{\lambda}$ and the parameters, $\boldsymbol{\alpha}$, remain constant, respectively. This implies that at least one term in the summation (21) decreases. Therefore, at the *i*-th iteration alternating Lemma 1 and Lemma 2, one always has: $D_{\mathrm{M}}^{(i)} < D_{\mathrm{M}}^{(i-1)}$, and the process necessarily minimizes cost (21). Theorem 1 proves that that cost is convex, hence the minimum is global, and the sequence $\{ D_M^{(1)}, D_M^{(2)}, D_M^{(3)}, ... D_M^{(n)} \}$ converges to the global minimum of (21).

VQ-SVM Optimization Algorithm for the Modified Dual

1. $\boldsymbol{\alpha} = \arg\min D_{M,SVM}, \ \ \lambda = 0$
2. $\tilde{\lambda} = \arg\min D_{M,a}$ as per (26)
3. Repeat until $\left| \left\| \mathbf{w} - \mathbf{w}^{(\mathrm{KM})} \right\|^2 - \rho_0^2 \right| < \tau$
   a. $\tilde{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} D_{\mathrm{M}}(\tilde{\lambda})$ as per (27)
   b. $\tilde{\lambda} = \arg\min_{\lambda} D_{\mathrm{M},\lambda}(\tilde{\boldsymbol{\alpha}})$ as per (26)

In the pseudocode, τ is a tolerance threshold to detect when the solution is close enough to the surface of the hypersphere.

B. Operational Aspects in the Optimization Procedure

The most computation-intensive phases of the above algorithm are the optimization processes at step 3.a) and step 3.b). It has been proved [21] that the computational cost of SMO is roughly a quadratic function of the number of patterns; the computational cost of (26) 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:

$$O\left((n_{SV})^2 + k \cdot np^2 \right) \tag{28}$$

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 $\tau = 10^{-3}$ 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 [1] 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 (19), and is written as:

$$sign(\mathbf{w}\cdot\Phi(\mathbf{x}_j)+b) = \\ = sign\left(\frac{\sum_i \alpha_i y_i \Phi(\mathbf{x}_i) + \lambda \mathbf{w}^{(\mathrm{KM})}}{(1+\lambda)}\cdot\Phi(\mathbf{x}_j)+b\right) \quad (29)$$

or equivalently:

$$sign(\mathbf{w}\cdot\Phi(\mathbf{x}_j)+b) = \\ = sign\left(\frac{1}{1+\lambda}\left(\sum_i \alpha_i y_i k_{ij} + \lambda \sum_i \alpha_i^{(\mathrm{KM})} y_i^{(\mathrm{KM})} k_{ij}\right)+b\right) \quad (30)$$

When $\lambda = 0$, the regularization is no more biased and the problem reduces to the classical SVM decision function (3). The bias term in eq.(30) obeys the following Lemma:

Lemma 3 : Let $\hat{b}$ be the Lagrange multiplier derived from minimizing problem (21): then the bias b appearing in the decision function (30) is $b = \hat{b}/(1+\lambda)$.

Proof: for clarity, the proof is given in the Appendix.

C. 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 (16) 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 that those provided by classical Structural Risk Minimization.

V. Experimental Results

The empirical validation of the proposed method involved extensive experiments on real-world datasets, namely, the MNIST numerical recognition testbed [12], text-classification problems drawn from the "Newsgroup-20" dataset [13]; reference UCI datasets "Heart", "Ionosphere", "Sonar", and "Pima Indian Diabetes" [13] were also tested.

A. Experimental Procedure

A version of SMO with RBF kernel supported the model presented in [22] 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 $\tau = 10^{-3}$. The model-selection approach scanned wide ranges of hyper-parameter settings; this lead 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 as per (8);
- 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, $\pi^{(\mathrm{TG})}$, that resulted from summing the training SVM error, ν, with the conventional MD-based penalty term computed as per (8);
- the generalization error bound, $\pi^{(\mathrm{VQ\text{-}TG})}$, that resulted from summing the training SVM error, ν, with the MD-based penalty term subject to the quadratic constraint (16);
- the "true" empirical generalization error, π_{VAL}, measured on the validation set, Z_{VAL}.

B. 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 $np = 200$. Each row in Table I addresses one of the 45 binary OCR problems and gives:

- the most promising hyper-parameters, $(C, 2\sigma^2)$, resulting from the model-selection process;
- the associate generalization bounds, for both the conventional and the constrained MD approach;

Experimental procedure

Input: training set Z, validation set Z_{VAL}, hyperparameters $\{C, \sigma\}$, sample size *NumIter*
Output: generalization predictions and estimates: $\pi^{(TG)}, \pi^{(VQ\text{-}TG)}, \pi_{VAL}$

1. Build the *kernel matrix*, **K**, on training data, Z.
2. *Train an SVM* on training data (original classes) → $\{\mathbf{w}^{(TG)}, b\}$ – ref. (2)
3. Compute the *empirical training error*, $\nu^{(TG)}$, using **y**, **K**, $\{\mathbf{w}^{(TG)}, b\}$ – ref (3)
4. Perform *unsupervised clustering* of unlabeled data $\{\mathbf{m}, \Psi_0, \Psi_1\}$
5. Apply an *artificial labeling* schema: $y^{(KM)+} \equiv \{\Psi_0 \to +1, \Psi_1 \to -1\}$

   Train an SVM on training data (artificial classes $y^{(KM)+}$) → $\{\mathbf{w}^{(KM)\pm}, b^{(KM)\pm}\}$ – ref. (10,11)
6. Select *unsupervised reference,* $\{\mathbf{w}^{(KM)}, y^{(KM)}, b^{(KM)}\}$ – ref (12)
7. *Maximal-Discrepancy generalization bounds* (conventional and modified SVM models)

   Set: $\bar{\nu}^{(MD)} := 0;\ \bar{\nu}^{(VQ-MD)} := 0;$

   For i = 1 to NumIter

   a. Apply *random-swap labeling* schema → $y^{(MD)}$ – ref (6)

   b. *Train an SVM* on training data using classes $y^{(MD)}$ → $\{\mathbf{w}^{(MD)}, b^{(MD)}\}$ – ref (2)

   c. Compute MD-*training error*, $\bar{\nu}_i$, using **K**, $\{\mathbf{w}^{(MD)}, b^{(MD)}\}, y^{(MD)}$ – ref (3)

   d. $\bar{\nu}^{(MD)} := \bar{\nu}^{(MD)} + \bar{\nu}_i$

   e. *Train a* constrained *SVM* on Z with classes $y^{(MD)}$ → $\{\mathbf{w}^{(VQ-MD)}, b^{(VQ-MD)}\}$ – ref (21)

   f. Compute the MD-*training error*, $\bar{\nu}_i^{(VQ)}$, $\{\mathbf{w}^{(VQ-MD)}, b^{(VQ-MD)}\}, y^{(MD)}$ – ref (30)

   g. $\bar{\nu}^{(VQ-MD)} := \bar{\nu}^{(VQ-MD)} + \bar{\nu}_i^{(VQ)}$
8. $\bar{\nu}^{(MD)} := \bar{\nu}^{(MD)} / NumIter;\quad \bar{\nu}^{(VQ-MD)} := \bar{\nu}^{(VQ-MD)} / NumIter$
9. Compute *generalization bound* (conventional SVM): $\pi^{(TG)} = \nu^{(TG)} + \left(1 - 2\bar{\nu}^{(MD)}\right)$
10. Compute *generalization bound* (constrained SVM): $\pi^{(VQ\text{-}TG)} = \nu^{(TG)} + \left(1 - 2\bar{\nu}^{(VQ\text{-}MD)}\right)$
11. Evaluate the *validation error* (conventional SVM), π_{VAL} on Z_{VAL} using **K**, $\{\mathbf{w}^{(TG)}, b\}$ – ref (3)

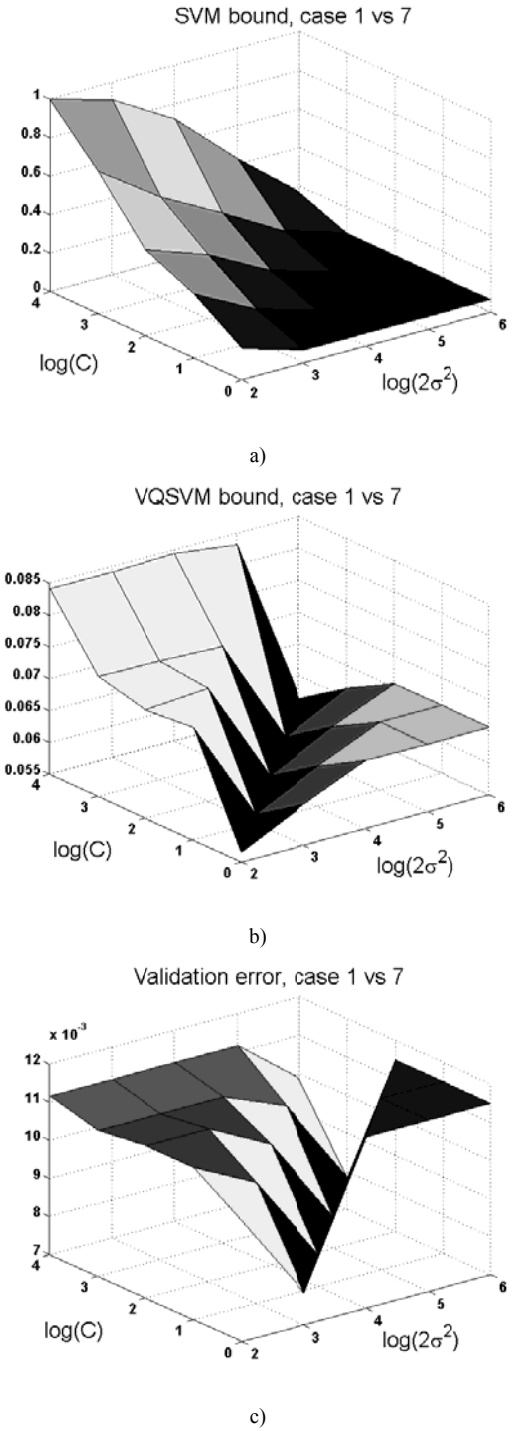


Fig.4 Model selection surfaces in the hyper-parameter space: a) Conventional-SVM MD-bound surface . b) Constrained-SVM MD bound surface c) Validation-error surface

- the validation error, $\pi_{VAL}^{(TG)}$, measured when using the model selection suggested by the conventional MD bound;

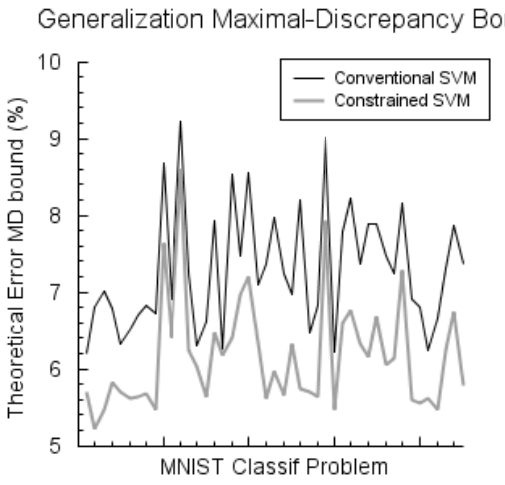
the validation error, $\pi_{VAL}^{(VQ\text{-}TG)}$, measured when using the model selection suggested by the constrained-SVM MD bound.

The constrained approach almost always yielded tighter generalization bounds. Even more importantly, the validation errors reduced accordingly, thus showing that the

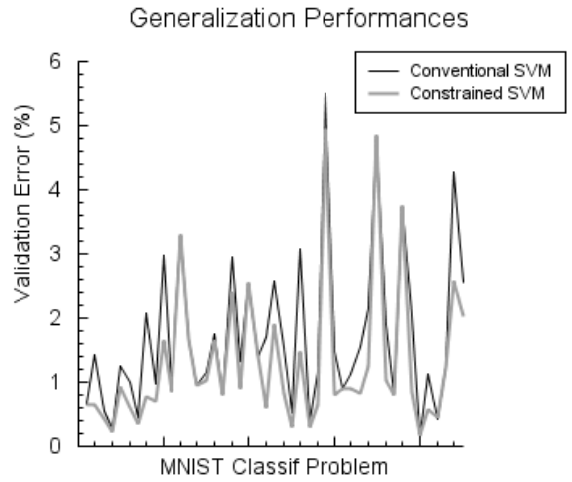
unsupervised reference also led to a better model selection. To demonstrate the different properties of either approach, Figure 4 gives the progression of the generalization bound (in the hyper-parameter space) for the problem '1' *vs/* '7', which proved quite difficult due to the similar appearances of the digits. The graphs clearly show that the constrained approach 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. The effectiveness of the constrained-SVM method can also be assessed by analyzing the results of Table I in a graphical way. Figure 5a compares the theoretical generalization bounds, one derived with a conventional Maximal-Discrepancy procedure (i.e using (2)), the other obtained by including the additional constraint (16). 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 (Fig.5b) 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.

A discussion about the general model-selection problem in supervised training is quite complex and is beyond the scope of this paper. 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 [23].



a)



b)

Fig.5 Comparison of conventional and constrained SVM model selection methods. a) Generalization bounds b) Validation error

TABLE I
NIST DIGIT RECOGNITION. MODEL-SELECTION RESULTS, BOUNDS, AND ACCURACY OF CONVENTIONAL SVM AND CONSTRAINED SVM.

Problem	Conventional SVM		Constrained SVM		Theoretical Bound		Validation Error	
	$C^{(\mathrm{TG})}$	$2\left[\sigma^{(\mathrm{TG})}\right]^2$	$C^{(\mathrm{VQ\text{-}TG})}$	$2\left[\sigma^{(\mathrm{VQ\text{-}TG})}\right]^2$	$\pi^{(\mathrm{TG})}$	$\pi^{(\mathrm{VQ\text{-}TG})}$	$\pi_{\mathrm{VAL}}^{(\mathrm{TG})}$	$\pi_{\mathrm{VAL}}^{(\mathrm{VQ\text{-}TG})}$
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%**

C. Newgroup-20 Experiments

This experimental campaign involved bi-class problems from the Newsgroup-20 dataset for text mining [13]. 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 [24] (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 [25] 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 [26]. 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 II 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 $C \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$, and $2\sigma^2 \in \{1, 10, 10^2, 10^3, 10^4, 10^5\}$.

TABLE II
NEWSGROUP-20 BINARY PROBLEMS

Dataset	#Training	#Test
sci.electronics VS *rec.sport.baseball*	200	1775
sci.space VS *sci.med*	200	1777
alt.atheism VS *sci.crypt*	200	1590
rec.sport.hockey VS *rec.sport.baseball*	200	1793
talk.politics.guns VS *talk.politics.mideast*	200	1650

The reported results confirmed some significant properties that have been 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.

D. Experiments on limited sample UCI datasets

The experimental verification of the VQSVM approach involved an additional set of reference datasets from UCI [13], 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 (np = 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: $C \in \{10^{-2}, 10^{-1}, 1, 10, 10^2\}$ $2\sigma^2 \in \{10^{-2}, 10^{-1}, 1, 10\}$. Tables IV (a,b,c), V (a,b,c) , VI (a,b,c), VII (a,b,c) 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" discussed in Section III.2 (Fig.1). 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.

TABLE III
NEWSGROUP-20 RESULTS

Problem	Conventional SVM		Constrained SVM		Theoretical Bound		Validation Error	
	$C^{(\mathrm{TG})}$	$2\left[\sigma^{(\mathrm{TG})}\right]^2$	$C^{(\mathrm{VQ\text{-}TG})}$	$2\left[\sigma^{(\mathrm{VQ\text{-}TG})}\right]^2$	$\pi^{(\mathrm{TG})}$	$\pi^{(\mathrm{VQ\text{-}TG})}$	$\pi_{\mathrm{VAL}}^{(\mathrm{TG})}$	$\pi_{\mathrm{VAL}}^{(\mathrm{VQ\text{-}TG})}$
sci.electronics rec.sport.baseball	1	1	1000	10	24.0%	6.3%	24.0%	**5.9%**
sci.space sci.med	1	1	1000	10	8.6%	6.4%	8.1%	**4.9%**
alt.atheism sci.crypt	10	1000	10	1	16.2%	5.4%	12.5%	**4.1%**
rec.sport.hockey, rec.sport.baseball	10	1	1000	10	34.0%	7.8%	7.4%	**7.2%**
talk.politics.guns, talk.politics.mideast	10	1	1000	10	16.89%	5.3%	4.24%	**3.64%**

TABLE IV A
"HEART" TESTBED. COMPARISON BETWEEN CLASSICAL MD AND VQSVM GENERALIZATION BOUNDS.

$2\sigma^2$	C									
	0.01		0.1		1		10		100	
	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$
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 IV B
"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 V A
"IONOSPHERE" TESTBED. COMPARISON BETWEEN CLASSICAL MD AND VQSVM GENERALIZATION BOUNDS.

$2\sigma^2$	C									
	0.01		0.1		1		10		100	
	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$
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 V B
"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 VI
"SONAR" TESTBED. COMPARISON BETWEEN CLASSICAL MD AND VQSVM GENERALIZATION BOUNDS.

$2\sigma^2$	C									
	0.01		0.1		1		10		100	
	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$
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 VII A
"DIABETES" TESTBED. COMPARISON BETWEEN CLASSICAL MD AND VQSVM GENERALIZATION BOUNDS.

$2\sigma^2$	C									
	0.01		0.1		1		10		100	
	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$	$\pi^{(TG)}$	$\pi^{(VQ\text{-}TG)}$
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 VII B
"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%

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.

VI. 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 [23].

The Maximum-Discrepancy (MD) probabilistic method for assessing a classifier's generalization ability features a sound theoretical background and can provide analytical bounds [16]. 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 the paper 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 paper 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 paper, 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 be extended 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 in this paper mainly aimed to set a starting point on the refinement of SRM, by using biased regularization for the computation of generalization bounds.

Future research may investigate to analyze the performances of the SVM model when using other constraints and reference models than Vector Quantization. Likewise, it is being studied the effects on the generalization bounds of applying biased regularization to various learning machines. Another point that is being studied is the effect on bounds when the ray of the hypersphere ρ_0^2 is a free parameter and not locked as in this theory; this means studying VQSVM as a learning tool and no more only as a support tool for SVM tight bounds computation.

APPENDIX

Proof of Theorem 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 $q_{ij}=y_iy_jK(\mathbf{x}_i,\mathbf{x}_j)$, and $\mathbf{G}$ the matrix of elements $g_{ij}=y_j\Phi(\mathbf{x}_j)_i$

Computing the second derivatives of eq.(21) leads to:

$$\mathbf{H}=\begin{pmatrix}\dfrac{1}{1+\lambda}\mathbf{Q} & \dfrac{-\mathbf{Q}\boldsymbol{\alpha}+\mathbf{w}^{(\mathrm{KM})}\mathbf{G}}{(1+\lambda)^2}\\ \left(\dfrac{-\mathbf{Q}\boldsymbol{\alpha}+\mathbf{w}^{(\mathrm{KM})}\mathbf{G}}{(1+\lambda)^2}\right)^{\mathrm{T}} & \dfrac{\|\mathbf{w}^{(\lambda 0)}-\mathbf{w}^{(\mathrm{KM})}\|^2}{(1+\lambda)^3}\end{pmatrix}$$

By explicit computation one obtains:

$$(\boldsymbol{\alpha}\ \lambda)\mathbf{H}\begin{pmatrix}\boldsymbol{\alpha}\\ \lambda\end{pmatrix}=\frac{\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}}{1+\lambda}+\frac{\lambda^2\|\mathbf{w}^{(\lambda 0)}-\mathbf{w}^{(\mathrm{KM})}\|^2}{(1+\lambda)^3}+$$
$$+\frac{2\lambda}{(1+\lambda)^2}\left(\left[\mathbf{w}^{(\lambda 0)}\right]^{\mathrm{T}}\mathbf{w}^{(\mathrm{KM})}-\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}\right)=$$
$$=$$
$$=\frac{\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}}{1+\lambda}+\frac{\lambda^2}{(1+\lambda)^3}\left(\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}-2\left[\mathbf{w}^{(\lambda 0)}\right]^{\mathrm{T}}\mathbf{w}^{(\mathrm{KM})}+\|\mathbf{w}^{(\mathrm{KM})}\|^2\right)+$$
$$+\frac{2\lambda}{(1+\lambda)^2}\left(\left[\mathbf{w}^{(\lambda 0)}\right]^{\mathrm{T}}\mathbf{w}^{(\mathrm{KM})}-\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}\right)=$$
$$=\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}\left(\frac{1}{1+\lambda}+\frac{\lambda^2}{(1+\lambda)^3}-\frac{2\lambda}{(1+\lambda)^2}\right)+$$
$$+\left[\mathbf{w}^{(\lambda 0)}\right]^{\mathrm{T}}\mathbf{w}^{(\mathrm{KM})}\left(\frac{2\lambda}{(1+\lambda)^2}-\frac{2\lambda^2}{(1+\lambda)^3}\right)+\frac{\lambda^2\|\mathbf{w}^{(\mathrm{KM})}\|^2}{(1+\lambda)^3}=$$
$$=\frac{1}{(1+\lambda)^3}\left(\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}+2\lambda\left[\mathbf{w}^{(\lambda 0)}\right]^{\mathrm{T}}\mathbf{w}^{(\mathrm{KM})}+\lambda^2\|\mathbf{w}^{(\mathrm{KM})}\|^2\right)$$

The term on brackets is $\|\mathbf{w}^{(\lambda 0)}+\lambda\mathbf{w}^{(\mathrm{KM})}\|^2$. 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. **.**

Proof of Lemma 1:

In this case it is necessary to compute λ that minimizes (23). By computing the derivative along λ and setting to 0 one gets: $\frac{\rho_0^2}{2}-\frac{\|\mathbf{w}^{(\lambda 0)}-\mathbf{w}^{(\mathrm{KM})}\|^2}{2(1+\lambda)^2}-\delta=0$. From which: $(1+\lambda)^2=\frac{\|\mathbf{w}^{(\lambda 0)}-\mathbf{w}^{(\mathrm{KM})}\|^2}{\rho_0^2-2\delta}$. 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: $(1+\tilde{\lambda})^2=\frac{\|\mathbf{w}^{(\lambda 0)}-\mathbf{w}^{(\mathrm{KM})}\|^2}{\rho_0^2}$. Between the two possible solutions, that with $+\sqrt{\frac{\|\mathbf{w}^{(\lambda 0)}-\mathbf{w}^{(\mathrm{KM})}\|^2}{\rho_0^2}}$ must be chosen being $1+\tilde{\lambda}>0$ (because $\lambda>0$). More explicitly $\tilde{\lambda}=\sqrt{\frac{\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}+\|\mathbf{w}^{(\mathrm{KM})}\|^2-2\left[\mathbf{w}^{(\lambda 0)}\right]^{\mathrm{T}}\mathbf{w}^{(\mathrm{KM})}}{\rho_0^2}}-1$. Now $\|\mathbf{w}^{(\mathrm{KM})}\|^2=\sum_{i,j=1}^{np}\alpha_i^{(\mathrm{KM})}\alpha_j^{(\mathrm{KM})}y_i^{(\mathrm{KM})}y_j^{(\mathrm{KM})}k(\mathbf{x_i},\mathbf{x_j})$ and $\left[\mathbf{w}^{(\lambda 0)}\right]^{\mathrm{T}}\mathbf{w}^{(\mathrm{KM})}=\sum_{i,j=1}^{np}\alpha_i\alpha_j^{(\mathrm{KM})}y_iy_j^{(\mathrm{KM})}k(\mathbf{x_i},\mathbf{x_j})$. Plugging all together the explicit formula for computing $\tilde{\lambda}$ is:

$$\tilde{\lambda}=\left(\left[\sum_{i,j=1}^{np}\alpha_i\alpha_jy_iy_jk(\mathbf{x_i},\mathbf{x_j})+\right.\right.$$
$$+\sum_{i,j=1}^{np}\alpha_i^{(\mathrm{KM})}\alpha_j^{(\mathrm{KM})}y_i^{(\mathrm{KM})}y_j^{(\mathrm{KM})}k(\mathbf{x_i},\mathbf{x_j})+$$
$$\left.\left.-2\sum_{i,j=1}^{np}\alpha_i\alpha_j^{(\mathrm{KM})}y_iy_j^{(\mathrm{KM})}k(\mathbf{x_i},\mathbf{x_j})\right]\cdot\rho_0^{-2}\right)^{1/2}-1$$

Proof of Lemma 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 $\left(\frac{\|\mathbf{w}^{(\lambda 0)}\|^2}{2}-\sum_i\alpha_i\right)-\frac{\lambda}{2(1+\lambda)}(\boldsymbol{\alpha}'\mathbf{Q}\boldsymbol{\alpha}-2\boldsymbol{\alpha}'\boldsymbol{\beta})$ where $\boldsymbol{\beta}$ is the vector of elements $\beta_i=y_i\sum_j\alpha_j^{(\mathrm{KM})}y_j^{(\mathrm{KM})}k(\mathbf{x_i},\mathbf{x_j})$.

Moreover:

$$\left(\frac{\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}}{2}-\sum_i\alpha_i\right)-\frac{\lambda}{2(1+\lambda)}(\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}-2\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\beta})=$$
$$=\frac{1}{2(1+\lambda)}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}-\sum_i\alpha_i\left(1-\frac{\lambda}{(1+\lambda)}\beta_i\right)$$

or, multiplying by $(1+\lambda)$ in a more usual form as:

$$\frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha} - \sum_i \alpha_i(1+\lambda(1-\beta_i))$$

Proof of Lemma 3:

The Lagrange multiplier $\hat{b}$ obtained by the minimization process of (27) must be divided by the factor $1+\lambda$ to get the bias *b* to be used in the decision function (30). This aspect can be simply seen by looking at (23) where for obtaining the final cost (27) 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 $1+\lambda$.

ACKNOWLEDGMENT

The authors gratefully acknowledge the contribution by Dr. Fabio Rivieccio, for his preliminary studies on this research, and the fruitful, constructive suggestions by the anonymous reviewers.

REFERENCES

[1] V. Vapnik, Statistical Learning Theory. Wiley-Interscience Pub., New York, 1998.
[2] C. Cortes, V. Vapnik, "Support vector networks," Mach. Learn., vol. 20, pp. 273-297, 1995.
[3] P.L. Bartlett, S. Mendelson, "Rademacher and Gaussian complexities: risk bounds and structural results," J. Machine Learn. Res., vol. 3, pp. 463-482, 2002.
[4] V . Vapnik, E. Levin, Y. Le Cun, "Measuring the VC-Dimension of a Learning Machine," Neural Comput., vol. 6, pp. 851-876, 1994.
[5] R. Kohavi, "A study of cross validation and bootstrap for accuracy estimation and model selection," in Proc. IJCAI 1995.
[6] P. Bartlett, S. Boucheron, G. Lugosi, "Model selection and error estimation" Machine Learn., vol. 48, pp.85-113, 2001.
[7] O. Chapelle and B. Scholkopf and A. Zien, Semi-Supervised Learning, MIT Press, Cambridge, MA, 2006, http://www.kyb.tuebingen.mpg.de/ssl-book.
[8] X. Zhu, "Semi-Supervised Learning Literature Survey", Technical Report 1530, 2008 University of Wisconsin–Madison, http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf.
[9] V. Castelli, T.M. Cover, "The relative value of labeled and unlabeled samples in Pattern Recognition with an unknown mixing parameter", IEEE Trans Inform Theory, vol. 42(6), pp.2102-2117, 1996.
[10] N.B. Karayiannis, MI G. Weiqun, "Growing Radial Basis Neural Networks: merging supervised and unsupervised learning with network growth techniques," IEEE Trans. Neural Netw., vol.8(6), pp.1492-1506, Nov., 1997.
[11] S. Ridella, S. Rovetta, R. Zunino "K-Winner Machines for pattern classification," IEEE Trans Neural Netw, vol.12(2), pp. 371-385, Mar., 2001.
[12] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: "Gradient-Based Learning Applied to Document Recognition," Proc. IEEE, 86(11), pp.2278-2324, Nov., 1998. http://yann.lecun.com/exdb/mnist/
[13] S. Hettich, and S. D. Bay. The UCI KDD Archive [http://kdd.ics.uci.edu]. University of California, Dept. Inf. and Comput. Sci., 1999, Irvine, CA.
[14] http://www.clopinet.com/isabelle/Projects/SVM/applist.html
[15] K. Duan, S. Keerthi, A. Poo "Evaluation of simple performance measures for tuning svm hyperparameters," Neurocomputing, vol. 51, p.41-59, 2003.
[16] D. Anguita, S. Ridella, F. Rivieccio, R. Zunino, "Hyperparameter Design Criteria for Support Vector Classifiers" Neurocomputing , Special Issue on Support Vector Machines, 2003.
[17] B. Schölkopf and A. J. Smola, Learning with kernels, MIT Press, Cambridge MA, December 2001.
[18] S.P. Lloyd "Least squares quantization in PCM," IEEE Trans. Inf. Theory, vol. 28(1), pp.129-135, Mar., 1982.
[19] M. Girolami, "Mercer Kernel-Based Clustering in Feature Space", IEEE Trans. Neural Netw., vol.13(3), pp.780-784, Jun., 2002.
[20] J. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines", In B. Schölkopf, C. Burges, and A. Smola (eds.) Advances in Kernel Methods Support Vector Learning, MIT Press, 1999.
[21] T. Joachims, "Making Large-Scale SVM Learning Practical". In B. Schölkopf, C. Burges, and A. Smola (eds.) Advances in Kernel Methods Support Vector Learning, MIT Press, Cambridge MA, 1999.
[22] C.C. Chang, C.J. Lin, "LIBSVM : A Library for Support Vector Machines", 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
[23] D. Anguita, A. Boni, S. Ridella, "Evaluating the Generalization Ability of Support Vector Machines through the Bootstrap", Neural Processing Letters, vol. 11(1), pp.51-58, 2000.
[24] M. F. Porter, "An algorithm for suffix stripping," Program, Morgan Kaufmann Publishers Inc., vol. 14(3), pp. 130-137, 1980.
[25] G. Salton, A. Wong, and L.S. Yang, "A vector space model for information retrieval," Journal Amer. Soc. Inform. Sci., vol. 18, pp. 613-620
[26] S. Decherchi, P. Gastaldo, R. Zunino "K-Means clustering for Content Based Document Management in Intelligence", in Advances In Artificial Intelligence for Privacy Protection and Security, Editors: Augusti Solanas and Antoni Martinez Bellesté, World Scientific Publishing, 2009.

Sergio Decherchi (born Genoa, Italy, 1983) obtained the "Laurea" degree summa cum laude in Electronic Engineering in 2007 from Genoa University, Italy. Since 2005 he started collaborating with the Department of Biophysical and Electronics Engineering of Genoa University, where he is pursuing a PhD in Electronic Engineering and Computer Science on Machine Learning. His main research areas include: theoretical aspects of Machine Learning, large scale learning algorithms development, semi-supervised learning, dedicated hardware for learning machines and Text Mining.

Sandro Ridella received the "Laurea" degree in electronic engineering from the University of Genova, Italy, in 1966.
He is a full Professor in the Department of Biophysical and Electronic Engineering, University of Genova, Italy, where he teaches circuits and algorithms for signal processing. In the last five years, his scientific activity has been mainly focused on the field of neural networks.

Rodolfo Zunino (born Genoa, Italy, 1961) obtained the "Laurea" degree cum laude in Electronic Engineering from Genoa University in 1985. From 1986 to 1995 he was a research consultant with the Department of Biophysical and Electronic Engineering (DIBE) of Genoa University.
He is currently with DIBE as an Associate Professor, teaching Electronics for Embedded Systems and Electronics for Security. His main scientific interests include intelligent systems for Computer Security, network security and Critical Infrastructure Protection, embedded electronic systems for neural networks, efficient models for data representation and learning, massive-scale text-mining and text-clustering methods, and advanced techniques for multimedia data processing.
Rodolfo Zunino coauthored more than 170 scientific papers in International Journals and Conferences; he has been the Co-Chairman of the two Editions of the International Workshop on Computational Intelligence for Security in Information Systems (CISIS'08 and CISIS'09).
Since 2001 he is contributing as Associate Editor of the IEEE Transactions on Neural Networks, and has participated in the Scientific Committees of several International Events (ICANN'02, ICANN'09, IWPAAMS2004,

IWPAAMS2005, Applied Computing 2006). Rodolfo Zunino is a Senior Member of IEEE (CIS – Computational Intelligence Society).

Paolo Gastaldo obtained the "Laurea" degree in Electronic Engineering and a PhD in Space Sciences and Engineering (2004), both from Genoa University, Italy. Since 2004 he is with the Department of Biophysical and Electronics Engineering of Genoa University, where he is the recipient of a research grant on Intelligent Systems for Visual Quality Estimation sponsored by Philips Research Labs – Eindhoven (NL). His main research area include innovative systems for visual signal understanding, neural network-based methods for nonlinear information processing, and DSP-based architectures for advanced signal interpretation, such as intelligent object tracking for video surveillance and cryptography.

Davide Anguita received the Laurea degree in electronic engineering in 1989 and the Ph.D. in computer science and electronic engineering from the University of Genova, Genoa, Italy, in 1993. After working as a Research Associate at the International Computer Science Institute, Berkeley, CA, on special-purpose processors for neurocomputing, he joined the Department of Biophysical and Electronic Engineering, University of Genova, where he is currently associate professor of smart electronic systems. His current research focuses on the theory and application of kernel methods and artificial neural networks.