Maximal Discrepancy for Support Vector Machines

Davide Anguita[1] and Alessandro Ghio[1] and Sandro Ridella[1]

1- University of Genova -Dept. of Biophysical and Electronic Engineering
Via Opera Pia 11A - I-16145 Genova - Italy

Abstract. Several theoretical methods have been developed in the past years to evaluate the generalization ability of a classifier: they provide extremely useful insights on the learning phenomena, but are not as effective in giving good generalization estimates in practice. We focus in this work on the application of the Maximal Discrepancy method to the Support Vector Machine for computing an upper bound of its generalization bias.

1 Introduction

A successful approach for estimating the generalization error of a learning machine relies on hold–out or cross–validation estimates, which can be obtained by removing some of the available samples from the training set and use them as an independent test set [1]. It is well–known, however, that this approach has several drawbacks in the small–sample setting, where reducing the size of training set could decrease the reliability of the learner [2].

When dealing with classification problems, several theoretical results have been proposed to bound the generalization error using only in–sample estimates [3, 4, 5]. Unfortunately, these bounds are too loose to be of any practical use or their application is unfeasible. For this reason, we propose in this paper a practical procedure to apply a rigorous in–sample method, the Maximal Discrepancy [5], to a very well–known learning algorithm, the Support Vector Machine (SVM) [3, 6].

2 The Maximal Discrepancy of a Classifier

Let $D_l = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_l, y_l)\}$ be a set of i.i.d. patterns, with $\boldsymbol{x}_i \in \mathbb{R}^n$ and $y_i \in \mathcal{Y} = \{-1, +1\}$, where the data are obtained from the unknown distribution $P(\boldsymbol{x}, y)$. A *prediction rule* is a function $f : \mathbb{R}^n \rightarrow \mathcal{Y}_f \subseteq \mathbb{R}$, selected from a set $\mathcal{F}$, which can be applied to D_l to compute its *empirical* error rate $\nu(f) = \frac{1}{l}\sum_{i=1}^{l} L\left(f(\boldsymbol{x}_i), y_i\right)$, where $L : \mathcal{Y}_f \times \mathcal{Y} \rightarrow [0, 1]$ is a suitable *loss function.*

Usually, in classification tasks, we are interested in a *hard* loss function, which counts the number of misclassified samples:

$$L_H\left(f(\boldsymbol{x}_i), y_i\right) = \begin{cases} 0 & \text{if} \quad y_i f\left(\boldsymbol{x}_i\right) > 0 \\ 1 & \text{if} \quad y_i f\left(\boldsymbol{x}_i\right) \leq 0. \end{cases} \tag{1}$$

Unfortunately, the use of a hard loss function makes the problem of finding the optimal f computationally hard. Therefore, the conventional SVM algorithm makes use of the well–known *hinge* loss $L_\xi\left(f(\boldsymbol{x}_i), y_i\right) = \left[1 - y_i f(\boldsymbol{x}_i)\right]_+$ [6], which is convex and Lipschitz continuous, so that the search for the optimal prediction rule is greatly simplified. This simplification, however, has a severe drawback, because the unboundness of the hinge loss complicates the problem of predicting the generalization ability of f [5]. We propose here to use a non–convex but Lipschitz continuous *soft* loss function:

$$L_S\left(f(\boldsymbol{x}_i), y_i\right) = \begin{cases} L_H\left(f(\boldsymbol{x}_i), y_i\right) & \text{if} \quad y_i f\left(\boldsymbol{x}_i\right) \leq -1 \\ L_\xi\left(f(\boldsymbol{x}_i), y_i\right)/2 & \text{if} \quad y_i f\left(\boldsymbol{x}_i\right) \geq -1 \end{cases}, \tag{2}$$

which can assume any value in the range $[0, 1]$. Differently from other proposals [7, 8, 9], L_S assigns a weight equal to 1/2 to the samples that are exactly on the separating surface and the extreme values 0 or 1 are assigned to the patterns that lie outside a *margin* $|y_i f(\boldsymbol{x}_i)| \geq 1$.

In order to predict the generalization ability of a classifier, we are interested in the *generalization error* of f, defined as $\pi(f) = \mathbb{E}_{(\boldsymbol{x},y)} L(f(\boldsymbol{x}), y)$, which, unfortunately, cannot be computed since we do not know $P(\boldsymbol{x}, y)$.

It is well–known that $\nu(f)$ usually underestimates $\pi(f)$: in particular, the function $f^* = \arg\min_{f\in\mathcal{F}} \nu(f)$, which minimizes the empirical error, is affected by a generalization bias $(\pi(f^*) - \nu(f^*))$. This bias can be studied by considering its supremum respect to the class of functions $\mathcal{F}$, $\sup_{f\in\mathcal{F}}[\pi(f) - \nu(f)]$, which is a random variable that depends on the data and the set of functions $\mathcal{F}$ [5] and can be analyzed through the *Maximal Discrepancy* (*MD*) method.

Let us split D_l in two halves and compute the two empirical errors:

$$\nu^{(1)}(f) = \frac{2}{l}\sum_{i=1}^{\frac{l}{2}} L\left(f(\boldsymbol{x}_i), y_i\right), \quad \nu^{(2)}(f) = \frac{2}{l}\sum_{i=\frac{l}{2}+1}^{l} L\left(f(\boldsymbol{x}_i), y_i\right), \tag{3}$$

then the Maximal Discrepancy is

$$MD = \max_{f\in\mathcal{F}}\left(\nu^{(1)}(f) - \nu^{(2)}(f)\right). \tag{4}$$

An upper bound of the generalization error in terms of MD [5] can be derived by using the following theorem. The complete proof is omitted due to space constraints: the proof is similar to that of Theorem 9 in [5], since the bound can be obtained applying the McDiarmid's inequality [10].

Theorem 1. *Given a dataset D_l, consisting in l patterns $\boldsymbol{x}_i \in \mathbb{R}^n$, given a class of functions $\mathcal{F}$ and a loss function $L(\cdot,\cdot) \in [0,1]$, the following procedure can be replicated m times: (a) randomly shuffle the samples in D_l to obtain $D_l^{(j)}$; (b) compute $MD^{(j)}$ for each replicate. Then*

$$\pi(f) \leq \nu(f) + \frac{1}{m}\sum_{j=1}^{m} MD^{(j)} + 3\sqrt{\frac{-\log\left(\frac{\delta}{2}\right)}{2l}} \tag{5}$$

holds with probability $1 - \delta$.

It is interesting to note that, for any loss function $L(\cdot,\cdot) \in [0,1]$ for which

$$L\left(f(\boldsymbol{x}_i), y_i\right) = 1 - L\left(f(\boldsymbol{x}_i), -y_i\right), \tag{6}$$

including the soft loss of Eq.(2), the value of MD can be computed by a conventional empirical minimization procedure. Let us define a new data set, $D'_l = \{(\boldsymbol{x}'_1, y'_1), ..., (\boldsymbol{x}'_l, y'_l)\}$, such that $(\boldsymbol{x}'_i, y'_i) = (\boldsymbol{x}_i, -y_i)$ if $i \leq \frac{l}{2}$ and $(\boldsymbol{x}'_i, y'_i) = (\boldsymbol{x}_i, y_i)$ if $i > \frac{l}{2}$, then it is easy to show that

$$MD = \max_{f\in\mathcal{F}} \left(\nu^{(1)}(f) - \nu^{(2)}(f)\right) = 1 - 2\left(\min_{f\in\mathcal{F}} \nu'(f)\right), \tag{7}$$

where $\nu'(f)$ is the empirical error obtained on D'_l.

We have to deal now with the non–convexity of L_S (as with L_H), since the optimization problem for finding the minimum of the empirical error becomes intractable and can be solved only in an approximate way [9], even for moderate l. If a solution to this problem is not found, then the application of the previous bound becomes unappealing in practice. We propose to use the bound of Eq. (5) by applying a *peeling* technique [11], which allows to obtain an upper bound of $\min_{f\in\mathcal{F}} \nu(f)$ and a lower bound of $\min_{f\in\mathcal{F}} \nu'(f)$, at the expense of a slight increase of its looseness.

3 The Application of MD to the SVM

For the sake of simplicity, we will focus here on the linear SVM

$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b \tag{8}$$

because the non–linear formulation can be easily obtained through the usual kernel trick [6]. Furthermore, we will use the following formulation, for finding the values of the weights, which is equivalent to the conventional one [3]:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \boldsymbol{e}^T\boldsymbol{\xi} \qquad\qquad (9)\\ & \|\boldsymbol{w}\|^2 \leq w_{MAX}^2 \\ & y_i\left(\boldsymbol{w}^T\boldsymbol{x} + b\right) \geq 1 - \xi_i \quad \forall i \in [1,\ldots,l] \\ & \xi_i \geq 0 \quad \forall i \in [1,\ldots,l] \end{aligned}$$

where $e_i = 1\ \forall i$ and ξ_i are the slack variables introduced to obtain the hinge loss. Even if, as in the conventional formulation, the unboundedness of the hinge loss prevents us to apply Theorem 1, this formulation allows us to easily control $\mathcal{F}$, defined as the set of functions for which $\|\boldsymbol{w}\|^2 \leq w_{MAX}^2$ and $b \in (-\infty, +\infty)$.

We define an alternative slack variable $\eta_i = \min(2, \xi_i)$, which is related to the soft loss L_S, since $L_S\left(f(\boldsymbol{x}_i), y_i\right) = \frac{\eta_i}{2}$. The new slack variable can be used in the SVM formulation in order to bound the loss function: the resulting optimization problem is not convex, therefore we apply the peeling procedure.

3.1 The Peeling Technique

It is easy to note that the values of η_i and ξ_i coincide for all the patterns $\boldsymbol{x}_i$ for which $y_i f(\boldsymbol{x}_i) \geq -1$: in this case $L_S = L_\xi/2$ and the loss function is bounded.

In general, however, some patterns will be characterized by $y_i f(\boldsymbol{x}_i) < -1$: they are critical for computing the error, since the L_S and L_ξ do not coincide, therefore we will define them *Critical Support Vectors* (*CSVs*).

Let $\mathcal{S} = \{1, ..., l\}$ be the set of indexes of the l patterns of the dataset, $\mathcal{S}_C$ the set of indexes of the CSVs and $\mathcal{S}_N = \mathcal{S} \setminus \mathcal{S}_C$ the set of indexes of the remaining patterns. Then, a lower bound of $\min_{f\in\mathcal{F}} \nu'(f)$ or, in other words, an upper bound of MD, can be found using the following theorem (proofs are omitted due to space constraints):

Theorem 2. *Let D_l be a dataset of l patterns and let us suppose to know the values η_i for each pattern in D_l. Then, given a class of functions $\mathcal{F}$ as defined in the previous section:*

$$\min_{f\in\mathcal{F}} \frac{1}{l} \sum_{i\in\mathcal{S}} \frac{\eta_i}{2} \geq \min_{f\in\mathcal{F}} \frac{1}{l} \sum_{k\in\mathcal{S}_N} \frac{\eta_k}{2} = \min_{f\in\mathcal{F}} \frac{1}{l} \sum_{k\in\mathcal{S}_N} \frac{\xi_k}{2} \tag{10}$$

Similarly, we can upper bound the error on the training set $\min_{f\in\mathcal{F}} \nu(f)$:

Theorem 3. *Let D_l be a dataset of l patterns and let us suppose to know the values η_i for each pattern in D_l. Then, given a class of functions $\mathcal{F}$ as defined in the previous section:*

$$\min_{f\in\mathcal{F}} \frac{1}{l} \sum_{i\in\mathcal{S}} \frac{\eta_i}{2} \leq \frac{|\mathcal{S}_C|}{l} + \min_{f\in\mathcal{F}} \frac{1}{l} \sum_{k\in\mathcal{S}_N} \frac{\eta_k}{2} = \frac{|\mathcal{S}_C|}{l} + \min_{f\in\mathcal{F}} \frac{1}{l} \sum_{k\in\mathcal{S}_N} \frac{\xi_k}{2}, \tag{11}$$

where $|\mathcal{S}_C|$ is the cardinality of the set $\mathcal{S}_C$.

In order to obtain the tightest bound, we should choose the set $\mathcal{S}_C$ with minimum cardinality, but this approach is obviously infeasible as it would require to examine all the possible combinations of samples. A possible solution is to consider one sample at the time: at first, the SVM learning problem is solved to identify the CSVs, then the CSV with the largest error or, in other words, the sample for which $y_i f(\boldsymbol{x}_i)$ is minimum, is deleted from the training set and the learning is repeated with the remaining samples. At the final step, the classifier will be trained on the set consisting of the remaining $|\mathcal{S}_N|$ patterns. The following Theorem provides a lower bound for $|\mathcal{S}_N|$ and guarantees that the peeling procedure ends with $|\mathcal{S}_N| > 0$:

Theorem 4. *The peeling technique described above ends with $|\mathcal{S}_N| \geq d_{VC}$, where d_{VC} is the Vapnik–Chervonenkis dimension of the classifier [3].*

The peeling procedure is obviously sub–optimal and could remove, at least in theory, a large number of CSVs, so making the bound on generalization error very loose. In practice, however, the number of CSVs is usually a tiny fraction of the training set and several replicates are used in order to improve the actual value of the MD term in the bound.

As a last remark, we show that our approach is consistent in computing MD:

Theorem 5. *Let D_l be a dataset of l patterns. Let us suppose to know the soft loss values η_i for each pattern in D_l. Then, given a class of functions $\mathcal{F}$,*

$$\frac{1}{l}\min_{f\in\mathcal{F}}\sum_{i\in\mathcal{S}_N}\frac{\xi_i}{2}\le\frac{1}{2}. \tag{12}$$

Therefore $\min_{f\in\mathcal{F}}\nu'(f)\le 1/2$ and $MD\ge 0$.

4 Experimental Results

The MD–based method is obviously targeted toward small–sample problems, where the use of a hold–out set for estimating the generalization ability of a classifier is not reasonable. We propose to select a real–world dataset, consisting of a large number of samples, and use only a small amount of the available data as training set, so that the remaining samples can be used as a test set to obtain a good error estimate $\hat{\pi}$.

We select the well–known MNIST dataset [12] consisting of 62000 images, representing the numbers from 0 to 9: in particular, we consider the 13074 patterns containing 0's and 1's that allow us to deal with a binary classification problem. We build the training set by randomly sampling a small number of patterns, varying from $l = 20$ to $l = 300$, while the remaining $13074-l$ images are used as a test set. Furthermore, in order to avoid unlucky training–test splittings and build statistically relevant results, we repeat each random sampling 30 times. Note that the dimensionality of the dataset is 784, which is much higher than the number of samples in each of the training sets and, therefore, defines a typical small–sample setting.

We apply the procedure described before to find an upper bound of $\nu(f)$ and a lower bound of $\nu'(f)$ for $f\in\mathcal{F}$ and substitute these values in Eq. (5), where $m = 30$.

The results obtained using our approach, which uses a bounded loss function, are detailed in column BL of Table 1 and are compared with an unbounded loss approach (UL), where the error estimate is computed with L_S *after* the learning phase without removing the CSVs. The results are shown in Table 1, where the term depending on δ is omitted as it is the same for all the cases. The last column is the test set error ($\hat{\pi}$), computed with L_S, which can be used as a good approximation of the true error. Our approach is the only one which guarantees that the MD value is never less than zero (thank to Theorem 5), differently from the case UL, where approximately 4% of the 30 replicates give an inconsistent value. The bounded loss estimate (BL) is comparable with the unbounded one (UL), but the first one is obviously to be preferred as it can be obtained with a relatively small effort.

l	BL	UL	$\hat{\pi}$
20	32.5 ± 1.2	34.1 ± 1.4	13.0 ± 0.8
50	22.2 ± 0.5	22.2 ± 0.5	8.2 ± 0.5
100	17.5 ± 0.5	17.3 ± 0.6	6.1 ± 0.5
200	14.5 ± 0.3	12.9 ± 0.4	4.4 ± 0.5
300	12.9 ± 0.3	10.5 ± 0.3	4.0 ± 0.4

Table 1: Generalization error estimates with 95% confidence intervals.

5 Conclusions

We detailed a procedure that allows us to estimate the in–sample generalization error of the SVM, which is particularly suitable for the small–sample setting. Much work will also be necessary to understand how to tighten the MD–based bounds; on the other hand, it is clear that data–dependent bounds are very promising tools. Our proposal, which allows to transfer from theory to practice the application of MD–based bounds to the SVM, is a first step toward a better understanding of this approach.

References

[1] A. Blum, A. Kalai, and J. Langford. Beating the hold–out: Bounds for k–fold and progressive cross–validation. In *Proc. of the Conference on Learning Theory (COLT)*, pages 203–208, 1999.

[2] A. Isaksson, M. Wallman, H. Goeransson, and M.G. Gustafsson. Cross–validation and bootstrapping are unreliable in small sample classification. *Pattern Recognition Letters*, 29:1960–1965, 2008.

[3] V.N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 2000.

[4] T. Poggio, R. Rifkin, S. Mukherjee, and P. Niyogi. General conditions for predictivity in learning theory. *Nature*, 428:419–422, 2004.

[5] P.L. Bartlett, S. Boucheron, and G. Lugosi. Model selection and error estimation. *Machine Learning*, 48:85–113, 2002.

[6] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines.* Cambridge University Press, 2000.

[7] L. Mason, J. Baxter, P.L. Bartlett, and M. Frean. Functional gradient techniques for combining hypotheses. In A. Smola, P.L. Bartlett, and B. Schoelkopf, editors, *Advances in Large Margin Classifiers.* The MIT Press, 2000.

[8] D. Anguita, S. Ridella, F. Rivieccio, and R. Zunino. Hyperparameter design criteria for training support vector machines. *Neurocomputing*, 55:109–134, 2003.

[9] L. Wang, H. Jia, and J. Li. Training robust support vector machines with smooth ramp loss in the primal space. *Neurocomputing*, 71:3020–3025, 2008.

[10] C. McDiarmid. On the method of bounded differences. In J. Siemons, editor, *Surveys in Combinatorics.* Cambridge University Press, 1989.

[11] M. Fadili, M. Melkemi, and A. ElMoataz. Non–convex onion–peeling using a shape hull algorithm. *Pattern Recognition Letters*, 25:1577–1585, 2004.

[12] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proc. of the International Conference on Machine Learning (ICML07)*, pages 473–480, 2007.