ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

Structural Risk Minimization and Rademacher Complexity for Regression

Davide Anguita, Alessandro Ghio, Luca Oneto and Sandro Ridella

University of Genova - Department of Biophysical and Electronic Engineering
Via Opera Pia 11A, I-16145 Genova - Italy

Abstract. The Structural Risk Minimization principle allows estimating the generalization ability of a learned hypothesis by measuring the complexity of the entire hypothesis class. Two of the most recent and effective complexity measures are the Rademacher Complexity and the Maximal Discrepancy, which have been applied to the derivation of generalization bounds for kernel classifiers. In this work, we extend their application to the regression framework.

1 Introduction

The Rademacher Complexity (RC) and the Maximal Discrepancy (MD) are two well-known data-dependent measures of complexity that have been deeply investigated in the last years [1, 2, 3, 4]. These measures have been exploited for deriving powerful statistical bounds on the performance of a learned model, as they provide sharper alternatives to both data–independent [5] and margin–based bounds [2]. For this reason, a lot of work has been spent in order to design new algorithms for effectively computing and exploiting these quantities in binary classification problems [6, 7]. In this paper, we propose the application of MD and RC to the regression framework [8, 9]: in particular, we derive a kernel algorithm, taking inspiration from the well–known Support Vector Regression (SVR) learning machine [10] and present some preliminary result on a simple artificial problem.

2 Complexity and Structural Risk Minimization

Let us consider the usual regression framework, consisting of an input set $\mathcal{X} \subseteq \mathbb{R}^d$ and an output set $\mathcal{Y} \subseteq \mathbb{R}$, which are related by a fixed but unknown probability distribution μ on $\mathcal{X} \times \mathcal{Y}$. A series of IID samples $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$, where $\boldsymbol{x}_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, is originated from μ and our goal is to find a regression function $h : \mathcal{X} \to \mathbb{R}$, chosen in a fixed class $\mathcal{H}$, along with a reliable estimate of its performance. For this purpose, a loss function $\ell\left(h(\boldsymbol{x}), y\right) : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$ is defined, which measures the quality of the regressor. Typical loss functions in the regression framework are the Mean Square Error (MSE), the Mean Absolute Error (MAE) or more sophisticated ones like the Huber's loss. Unfortunately, all the mentioned losses are unbounded, and this represents an issue for several statistical approaches, including both MD and RC [1, 2]. In particular, it is well–known that additional constraints on μ (e.g. on its high–order moments) must be introduced in this case, otherwise the convergence of the regressor to

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

the optimal one is not guaranteed [5, 11]. However, in practical cases, the data domain $(\mathcal{X}, \mathcal{Y})$ is always finite, so we can consider, without loss of generality, bounded losses of the type $\ell\left(h(\boldsymbol{x}), y\right) : \mathbb{R} \times \mathcal{Y} \rightarrow [0, 1]$.

Given a user–defined loss, the generalization error of a regression function h is defined as $L(h) = \mathbb{E}_{\mu}\ell\left(h(\boldsymbol{x}), y\right)$, which cannot be computed since μ is unknown. Exploiting its empirical estimate $\hat{L}_n(h) = \frac{1}{n}\sum_{i=1}^{n}\ell(h(\boldsymbol{x}_i), y_i)$ obviously brings to a severe over–fitting, therefore the Structural Risk Minimization (SRM) approach [2, 5, 1] suggests to study the uniform deviation $\sup_{h\in\mathcal{H}}[L(h) - \hat{L}_n(h)]$.

Let us define a complexity measure of the hypothesis space $\mathcal{H}$:

$$\hat{\mathcal{C}}_{\sigma}(\mathcal{H}) = \mathbb{E}_{\boldsymbol{\sigma}} \sup_{h\in\mathcal{H}} 2\hat{L}_n^{\boldsymbol{\sigma}} = -\mathbb{E}_{\boldsymbol{\sigma}} \inf_{h\in\mathcal{H}} -2\hat{L}_n^{\boldsymbol{\sigma}} \tag{1}$$

where $\hat{L}_n^{\boldsymbol{\sigma}} = \frac{1}{n}\sum_{i=1}^{n}\sigma_i\ell(h(\boldsymbol{x}_i), y_i)$ and $\boldsymbol{\sigma}$ is a vector of independent uniform $\{-1, +1\}$–valued random variables. The term defined in Eq. (1) is known as the Rademacher Complexity of the class $\mathcal{H}$, while, if the combinations of the random variables are such that $\sum_{i=1}^{n}\sigma_i = 0$, the Maximal Discrepancy is obtained, instead.

Given that the complexity measure is valid for any function $h \in \mathcal{H}$, it is possible to prove the following bound for $L(h)$ [3, 6], which holds with probability $(1 - \delta)$:

$$L(h) \leq \hat{L}_n(h) + \hat{\mathcal{C}}_{\sigma}(\mathcal{H}) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2n}}, \quad \forall h \in \mathcal{H}. \tag{2}$$

Eq. (2) can be used as a performance index in the SRM framework [5], by choosing a possibly infinite sequence $\{\mathcal{H}_i, i = 1, 2, \ldots\}$ of model classes of increasing complexity, $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \ldots$, and then selecting the class $\mathcal{H}^*$, and the function h^* in it, according to the best trade–off between the complexity $\hat{\mathcal{C}}_{\sigma}(\mathcal{H})$ and the empirical error $\hat{L}_n(h)$. Unfortunately, the term $\hat{\mathcal{C}}_{\sigma}(\mathcal{H})$ is difficult to compute in practice and, only in some recent works [6, 7] effective methods have been proposed, targeting Support Vector classifiers. In the following Section we extend them to Support Vector Regression.

3 RC and MD for Support Vector Regression

Let us define $h(\boldsymbol{x}) = \boldsymbol{w}\cdot\boldsymbol{\phi}(\boldsymbol{x}) + b$ as a linear regressor in $\boldsymbol{\phi}(\boldsymbol{x})$, where $\boldsymbol{\phi}(\boldsymbol{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^D$ is introduced as it will allow us to apply the well–known kernel trick [10], $\boldsymbol{w} \in \mathbb{R}^D$ and $b \in \mathbb{R}$. Let our class of functions consist in all the regressors for which $\|\boldsymbol{w}\|^2 \leq A$ and $b \in \mathbb{R}$ or, in other words, regressors with margin larger than $\frac{1}{A}$ [5]. As the epsilon insensitive loss function of SVR [10] $\ell_{\epsilon}(h(\boldsymbol{x}), y) = |h(\boldsymbol{x}) - y|_{\epsilon}$, where $|\cdot|_{\epsilon} = \max(0, |\cdot| - \epsilon)$, is unbounded, we introduce a bounded epsilon insensitive loss function $\ell_{\epsilon_l}^{\epsilon_u}(h(\boldsymbol{x}), y) = |h(\boldsymbol{x}) - y|_{\epsilon_l}^{\epsilon_u}$ (Figure 1), where $|\cdot|_{\epsilon_l}^{\epsilon_u} = \min\left(\max\left(0, |\cdot| - \epsilon_l\right), \epsilon_u\right)/\epsilon_u$, so that $\ell_{\epsilon_l}^{\epsilon_u}(h(\boldsymbol{x}), y) \in [0, 1]$. In order to identify h^* and $\mathcal{H}^*$, we have to both solve $\inf_{h\in\mathcal{H}}\hat{L}_n^{\sigma}$, and find the minimum of the empirical error. Note however, that the empirical error is already included

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

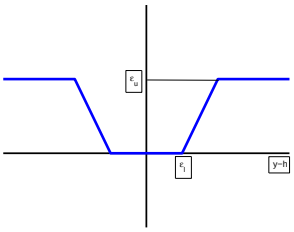


Fig. 1: The bounded epsilon insensitive loss function $\ell_{\epsilon_l}^{\epsilon_u}(h(\boldsymbol{x}), y)$.

in the computation, because $\hat{L}_n(h) = \hat{L}_n^{\boldsymbol{\sigma}}$ when $\sigma_i = -1$, $\forall i$. Then our problem can be reformulated as:

$$\inf_{\boldsymbol{w},b} \quad -\sum_{i=1}^{n} \sigma_i \ell_{\epsilon_l}^{\epsilon_u}(h(\boldsymbol{x}_i), y_i), \quad \text{s.t. } \|\boldsymbol{w}\|^2 \leq A, \tag{3}$$

which is a non–convex problem, based on Ivanov regularization. The corresponding Tikhonov regularization formulation becomes

$$\inf_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}(-\sigma_i)\ell_{\epsilon_l}^{\epsilon_u}(h(\boldsymbol{x}_i), y_i), \tag{4}$$

which is, in general, simpler to solve, though still non–convex. On the other hand, it is well–known that the two formulations are equivalent for some value of C, as shown, for example, in [12][1]. A solution to (4) can be found iteratively, by exploiting the ConCave–Convex Procedure (CCCP) [13, 14]. Though the global solution cannot be found, in general, because the problem is non–convex, the CCCP allows to reach a, usually good, local minimum in a finite number of steps [13]. In order to apply the CCCP, the concave and convex terms in (4) must be identified, therefore we define $\mathcal{S}^+$ as the set of indexes for which $-\sigma_i = +1$ and $\mathcal{S}^-$ as the set of indexes for which $-\sigma_i = -1$. Then, problem (4) can be reformulated as follows:

$$\min_{\substack{\boldsymbol{\theta} \\ \boldsymbol{\xi},\hat{\boldsymbol{\xi}} \\ \boldsymbol{\xi}',\hat{\boldsymbol{\xi}}'}} \overbrace{\frac{1}{2}\|\boldsymbol{w}\|^2}^{\mathcal{J}_{\text{convex}}(\boldsymbol{\theta})} + C\left(\sum_{i\in\mathcal{S}^+} \overbrace{\left(\xi_i + \hat{\xi}_i\right)}^{\mathcal{J}_{\text{convex}}(\boldsymbol{\theta})} \overbrace{- \left(\xi_i' + \hat{\xi}_i'\right)}^{\mathcal{J}_{\text{concave}}(\boldsymbol{\theta})} + \sum_{i\in\mathcal{S}^-} \overbrace{\left(\xi_i' + \hat{\xi}_i'\right)}^{\mathcal{J}_{\text{concave}}(\boldsymbol{\theta})} \overbrace{- \left(\xi_i + \hat{\xi}_i\right)}^{\mathcal{J}_{\text{convex}}(\boldsymbol{\theta})}\right)$$

$$\mathcal{S}^+ : \begin{cases} y_i - h_i \leq \epsilon_l + \xi_i & \xi_i \geq 0 \\ h_i - y_i \leq \epsilon_l + \hat{\xi}_i & \hat{\xi}_i \geq 0 \\ y_i - h_i \leq \epsilon_l + \epsilon_u + \xi_i' & \xi_i' \geq 0 \\ h_i - y_i \leq \epsilon_l + \epsilon_u + \hat{\xi}_i' & \hat{\xi}_i' \geq 0 \end{cases} \quad \mathcal{S}^- : \begin{cases} y_i - h_i \geq \epsilon_l + \xi_i & \xi_i \leq \epsilon_u \\ h_i - y_i \geq \epsilon_l + \hat{\xi}_i & \hat{\xi}_i \leq \epsilon_u \\ y_i - h_i \geq \epsilon_l + \xi_i' & \xi_i' \leq 0 \\ h_i - y_i \geq \epsilon_l + \hat{\xi}_i' & \hat{\xi}_i' \leq 0 \end{cases} \tag{5}$$

where, for simplifying the notation, $\boldsymbol{\theta} = [\boldsymbol{w}, b]$ and $h_i = h(\boldsymbol{x}_i) = \boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}_i) + b$. Then, we can apply the CCCP, as sketched in Algorithm 1.

[1]It is worth noting that the properties of interest, presented in [12], do not necessitate the convexity hypothesis.

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

At the t-th optimization step, the CCCP requires that the derivative of the concave part of the cost function is computed. Let $h_i^{(t)} = \boldsymbol{w}^{(t)} \cdot \boldsymbol{\phi}(\boldsymbol{x}_i) + b^{(t)}$ be the regressor, computed exploiting the solution identified at the t-th step. Then, we define:

$$\mathcal{S}^+ \left\{ \begin{array}{lll} \Delta_i & \left\{ \begin{array}{ll} +C & \text{If } \delta_i^{(t)} \geq \epsilon_l + \epsilon_u \\ 0 & \text{Otherwise} \end{array} \right. \\ \hat{\Delta}_i & \left\{ \begin{array}{ll} -C & \text{If } \delta_i^{(t)} \leq -\epsilon_l - \epsilon_u \\ 0 & \text{Otherwise} \end{array} \right. \end{array} \right. \quad \mathcal{S}^- \left\{ \begin{array}{lll} \Delta_i & \left\{ \begin{array}{ll} -C & \text{If } \delta_i^{(t)} \leq \epsilon_l \\ 0 & \text{Otherwise} \end{array} \right. \\ \hat{\Delta}_i & \left\{ \begin{array}{ll} +C & \text{If } \delta_i^{(t)} \geq -\epsilon_l \\ 0 & \text{Otherwise} \end{array} \right. \end{array} \right. \tag{6}$$

where $\delta_i^{(t)} = y_i - h_i^{(t)}$. Therefore, the derivative of the concave part is:

$$\left.\frac{d\mathcal{J}_{\text{concave}}(\boldsymbol{\theta})}{d\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}^{(t)}} \boldsymbol{\theta} = \sum_{i \in \mathcal{S}^+} \left(\Delta_i^{(t)} + \hat{\Delta}_i^{(t)}\right) h_i + \sum_{i \in \mathcal{S}^-} \left(\Delta_i^{(t)} + \hat{\Delta}_i^{(t)}\right) h_i \tag{7}$$

and the problem (5) at step t becomes:

$$\begin{aligned} \{\boldsymbol{w}^{(t+1)}, b^{(t+1)}\} : \arg \min_{\boldsymbol{w},b,\boldsymbol{\xi},\hat{\boldsymbol{\xi}}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C \sum_{i \in \mathcal{S}^+} \left(\xi_i + \hat{\xi}_i\right) - C \sum_{i \in \mathcal{S}^-} \left(\xi_i + \hat{\xi}_i\right) + \\ & + \sum_{i \in \mathcal{S}^+} \left(\Delta_i^{(t)} + \hat{\Delta}_i^{(t)}\right) \left(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i) + b\right) + \\ & + \sum_{i \in \mathcal{S}^-} \left(\Delta_i^{(t)} + \hat{\Delta}_i^{(t)}\right) \left(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i) + b\right) \end{aligned} \tag{8}$$

$$\mathcal{S}^+ : \left\{ \begin{array}{ll} y_i - \left(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i) + b\right) \leq \epsilon_l + \xi_i & \xi_i \geq 0 \\ \left(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i) + b\right) - y_i \leq \epsilon_l + \hat{\xi}_i & \hat{\xi}_i \geq 0 \end{array} \right.$$

$$\mathcal{S}^- : \left\{ \begin{array}{ll} y_i - \left(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i) + b\right) \geq \epsilon_l + \xi_i & \xi_i \leq \epsilon_u \\ \left(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i) + b\right) - y_i \geq \epsilon_l + \hat{\xi}_i & \hat{\xi}_i \leq \epsilon_u \end{array} \right.$$

By introducing $2n$ Lagrange multipliers $\boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}} \in \mathbb{R}^n$ and defining $Q \in \mathbb{R}^{n \times n} = \{q_{i,j}\} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\phi}(\boldsymbol{x}_i) \cdot \boldsymbol{\phi}(\boldsymbol{x}_j)$, where $K(\cdot,\cdot)$ is the kernel function, we can

Algorithm 1 The CCCP procedure.

Initialize $\boldsymbol{\theta}^0$
repeat
$\boldsymbol{\theta}^{(t+1)} = \arg\min_{\boldsymbol{\theta}} \mathcal{J}_{\text{convex}}(\boldsymbol{\theta}) + \left(\left.\frac{d\mathcal{J}_{\text{concave}}(\boldsymbol{\theta})}{d\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}^{(t)}} \right) \boldsymbol{\theta}$
until $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)}$

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

derive the dual formulation of problem (8) at the t-th step:

$$\begin{aligned}
\min_{\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}}} \quad & \frac{1}{2}\begin{bmatrix}\boldsymbol{\alpha}\\ \hat{\boldsymbol{\alpha}}\end{bmatrix}^T \begin{bmatrix}Q & Q\\ Q & Q\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}\\ \hat{\boldsymbol{\alpha}}\end{bmatrix} + \sum_{i\in\mathcal{S}^+} \alpha_i\left[-y_i+\epsilon_l\right] + \hat{\alpha}_i\left[-y_i-\epsilon_l\right] + \\
& + \sum_{i\in\mathcal{S}^-} \alpha_i\left[-y_i+\epsilon_l+\epsilon_u\right] + \hat{\alpha}_i\left[-y_i-\epsilon_l-\epsilon_u\right] \qquad (9)\\
& \sum_{i=1}^{n}\alpha_i + \sum_{i=1}^{n}\hat{\alpha}_i = 0\\
& \mathcal{S}^+ : \begin{cases} -\Delta_i^{(t)} \le \alpha_i \le C-\Delta_i^{(t)} \\ -C-\hat{\Delta}_i^{(t)} \le \hat{\alpha}_i \le -\hat{\Delta}_i^{(t)}\end{cases} \quad \mathcal{S}^- : \begin{cases} -C-\Delta_i^{(t)} \le \alpha_i \le -\Delta_i^{(t)} \\ -\hat{\Delta}_i^{(t)} \le \hat{\alpha}_i \le C-\hat{\Delta}_i^{(t)}\end{cases}
\end{aligned}$$

which can be solved with well-known Quadratic Programming solvers like, for example, SMO [9]. Once a solution has been found, the regressor is defined as $h(\boldsymbol{x}) = \sum_{i=1}^{n}(\alpha_i+\hat{\alpha}_i)K(\boldsymbol{x}_i,\boldsymbol{x}) + b$.

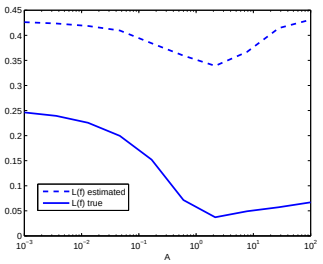
4 A simple example

We consider a simple regression problem, proposed in [14], where the function $g(x) = \mathrm{sinc}(3x)$ is uniformly sampled in $x \in [-1,+1]$ using 100 samples. An additive Gaussian noise $\mathcal{N}(0, 0.05)$ is applied to all samples, while a larger Gaussian noise $\mathcal{N}(0,1)$ is applied only to the 30% of the samples. For our experiments, we exploit a Gaussian kernel $\mathcal{K}(\boldsymbol{x}_i,\boldsymbol{x}_j) = \exp\left[-\gamma\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|^2\right]$, therefore the size of $\mathcal{H}$ is controlled by two hyperparameters: A and γ. In particular, we look for the optimal pair (A^*,γ^*), according to the SRM principle, by exploring the intervals $\gamma \in [10^{-3}, 10^2]$ and $A \in [10^{-3}, 10^2]$, which include the cases of interest, among 10 values, equally spaced in a logarithmic scale. We set $\epsilon_l = 0$, as in ℓ_1–regression, and $\epsilon_u = 1$, which allows to span the entire error range for the $\mathrm{sinc}(\cdot)$ function. In Fig. 2 the results, obtained using the RC, computed through a Monte Carlo procedure with 100 trials, are shown: similar values can be obtained with MD but are not presented here due to space constraints. Fig. 2a compares the trends of the Mean Absolute Percentage Error (MAPE) of $h^*(x)$ against $g(x)$ and of the error, predicted with the RC–based bound, when we set $\gamma = \gamma^*$ and we let A vary. It is worth noting that the minimum of the MAPE and of the predicted error coincide though, as usually happens with SRM bounds, the estimation is loose. This is confirmed also by Fig. 2b, which clearly shows the accuracy of the selected approximating function $h^*(x)$.

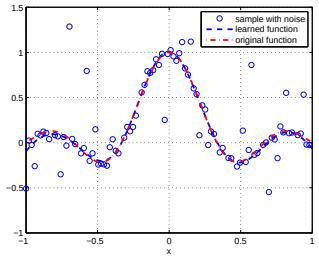
5 Conclusions

We have presented the application of the Rademacher Complexity and the Maximal Discrepancy to a bounded version of the Support Vector Regression. From the preliminary results on a simple artificial problem, it appears that these complexity measures can effectively identify the optimal regressor. More experiments are underway to compare this approach to the classical methods for measuring the goodness of fit of a statistical model, such as the one surveyed in [15].

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.



(a) Comparison of the MAPE of the selected regressor and the predicted error.



(b) Selected approximating function.

Fig. 2: Results obtained using the Rademacher Complexity approach.

References

[1] V. Koltchinskii and D. Panchenko. Rademacher processes and bounding the risk of function learning. *High Dimensional Probability II*, 47:443–459, 2000.

[2] P. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[3] D. Anguita, A. Ghio, L. Oneto, and S. Ridella. Maximal Discrepancy Vs. Rademacher Complexity for Error Estimation. In *Proc. of the European Symposium on Artificial Neural Networks*, 2011.

[4] D. Anguita, A. Ghio, L. Oneto, and S. Ridella. The Impact of Unlabeled Patterns in Rademacher Complexity Theory for Kernel Classifiers. In *Proc. of Neural Information Processing Systems*, 2011.

[5] V.N. Vapnik. *Statistical learning theory*. Wiley-Interscience, 1998.

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

[7] D. Anguita, A. Ghio, and S. Ridella. Maximal Discrepancy for Support Vector Machines. *Neurocomputing*, 74:1436–1443, 2011.

[8] G. Lugosi and K. Zeger. Nonparametric estimation via empirical risk minimization. *IEEE Transactions on Information Theory*, 41(3):677–687, 1995.

[9] B. Scholkopf, A.J. Smola, R.C. Williamson, and P.L. Bartlett. New support vector algorithms. *Neural computation*, 12(5):1207–1245, 2000.

[10] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines: and other kernel-based learning methods*. Cambridge University Press, 2000.

[11] S. Kutin. Extensions to mcdiarmid's inequality when differences are bounded with high probability. Technical report, TR-2002-04, University of Chicago, 2002.

[12] D. Anguita, A. Ghio, L. Oneto, and S. Ridella. In-sample Model Selection for Support Vector Machines. In *Proc. of the Int. Joint Conference on Neural Networks*, 2011.

[13] A.L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.

[14] Y.P. Zhao and J.G. Sun. Robust truncated support vector regression. *Expert Systems With Applications*, 37(7):5126–5133, 2010.

[15] V. Cherkassky and Y. Ma. Comparison of model selection for regression. *Neural computation*, 15(7):1691–1714, 2003.