ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

Local Rademacher Complexity Machine

Luca Oneto[1], Sandro Ridella[2], and Davide Anguita[1]

1 - DIBRIS - University of Genova
Via Opera Pia 13, I-16145 Genova - Italy

2 - DITEN - University of Genova
Via Opera Pia 11A, I-16145 Genova - Italy

Abstract. In this paper we present the Local Rademacher Complexity Machine, a transposition of the Local Rademacher Complexity Theory into a learning algorithm. By exploiting a series of real world small-sample datasets, we show the advantages of our proposal with respect to the Support Vector Machines, i.e. the transposition of the milestone results of V. N. Vapnik and A. Chervonenkis into a learning algorithm.

1 Introduction

Support Vector Machines (SVMs) are a state-of-the-art and powerful learning algorithm that can effectively solve many real world problems [1, 2]. SVMs are the transposition of the Vapnik-Chervonenkis (VC) Theory [3] into a learning algorithm which optimizes the trade-off between accuracy and complexity of the learned model [4].

VC Theory was later improved by the Rademacher Complexity (RC) Theory [5, 6], but the real breakthrough was made with the Local RC (LRC) Theory [7, 8]. In fact, LRC was able, for the first time, to avoid taking into account the functions with high empirical error when measuring the complexity of a learned model.

LRC is a quite powerful tool for getting a deeper understanding of the learning process. In fact, authors have shown that RC and LRC can be effectively used for Model Selection purposes in many different applications (e.g. small sample problems [9], resource limited models [10], graph kernel learning [11], and multiple kernel learning [12]). Nevertheless, to the best knowledge of the authors, no one has tried to develop a learning algorithm based on the the LRC Theory.

For this reason, inspired by the SVMs, we proposed in this paper the Local Rademacher Complexity Machine (LRCM), a transposition of the Local Rademacher Complexity Theory into a learning algorithm which improves, from a theoretical point of view, the properties of the original SVMs by including the new intuition behind LRC. In fact, our proposal is able, as the SVMs, to generate both linear and nonlinear models by exploiting the kernel trick [13]. Moreover, LRCM introduces a new regularization term which is able to penalize the function with small error on the available data but also small error on a random labeled sample, by implementing the same idea behind the LRC Theory.

Since RC and LRC have shown to be a good option for model selection purposes, particularly in the small-sample setting, we make use of several Human

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

Gene Expression datasets in order to test if the proposed LRCM improves over the state-of-the-art SVMs algorithms. Results show that LRCM is able to improve, in a statistical relevant way, the accuracy with respect to the SVMs in this setting.

2 Local Rademacher Complexity Theory

We consider the conventional binary classification problem [3]: based on a random observation of $\boldsymbol{x} \in \mathcal{X} \in \mathbb{R}^d$, one has to estimate $y \in \mathcal{Y} \subseteq \{\pm 1\}$ by choosing a suitable hypothesis $h : \mathcal{X} \to \hat{\mathcal{Y}}$, where $h \in \mathcal{H}$. A learning algorithm selects $h \in \mathcal{H}$ by exploiting a set of labeled samples $\mathcal{D}_n : \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)\}$. $\mathcal{D}_n$ consists of a sequence of independent samples distributed according to μ over $\mathcal{X} \times \mathcal{Y}$. The generalization error $L(h) = \mathbb{E}_{(\boldsymbol{x},y)} \ell(h(\boldsymbol{x}), y)$, associated to an hypothesis $h \in \mathcal{H}$, is defined through a loss function $\ell : \hat{\mathcal{Y}} \times \mathcal{Y} \to [0, 1]$. As μ is unknown, $L(h)$ cannot be explicitly computed, but we can compute the empirical error, namely the empirical estimator of the generalization error $\hat{L}_n^y(h) = {}^1\!/_n \sum_{i=1}^n \ell\left(h\left(\boldsymbol{x}_i\right), y_i\right)$.

The purpose of any learning procedure is to select a model h with small $L(h)$ and consequently to estimate it. Different alternatives exist in order to perform this task. The original milestone result was the VC-Theory [3, 14], which has been later improved by the RC-Theory [5, 6]. VC and RC Theories state that with probability $(1 - \delta)$ and $\forall h^* \in \mathcal{H}$

$$L(h^*) \leq \hat{L}_n^y(h^*) + C(\mathcal{H}) + \phi_1(n, \delta), \tag{1}$$

where $\phi_1(n, \delta)$ is a confidence term and $C(\mathcal{H})$ is the complexity of $\mathcal{H}$ measured with the VC-Dimension or the RC. SVMs, one of the state-of-the-art learning algorithms [2], are the transposition of Eq. (1) [3]. In fact SVMs search for the h which minimizes the trade-off between accuracy $\hat{L}_n^y(h^*)$ and complexity $C(\mathcal{H})$ or, in other words, the estimated generalization error. The real breakthrough, with respect to Eq. (1) was made with the LRC-Theory [7, 8] which was able to take into account just the function in $\mathcal{H}$ with small error for estimating the complexity of $\mathcal{H}$.

In order to present the LRC-Theory we have to define the RC [5] $\hat{R}_n(\mathcal{H}) = \sup_{h \in \mathcal{H}} {}^2\!/_n \sum_{i=1}^n \sigma_i \ell\left(h\left(\boldsymbol{x}_i\right), y_i\right)$, where $\sigma_1, \ldots, \sigma_n$ are n $\{\pm 1\}$-valued independent Rademacher random variables for which $\mathbb{P}(\sigma_i = +1) = \mathbb{P}(\sigma_i = -1) = {}^1\!/_2$ and the deterministic counterpart of $\hat{R}_n(\mathcal{H})$ is $R_n(\mathcal{H}) = \mathbb{E}_{\sigma_1,\cdots,\sigma_n} \mathbb{E}_{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n} \hat{R}_n(\mathcal{H})$. Note that, the empirical RC is usually defined as $\mathbb{E}_{\sigma_1,\cdots,\sigma_n} \hat{R}_n(\mathcal{H})$ in order to improve the constants in the generalization bounds. In this paper, instead, since we want to define a learning algorithm, the constants are not important. The important aspect, instead, is the computational requirement needed for computing the generalization bound and to keep it as limited as possible. Note also that, if $\ell(h(\boldsymbol{x}), y)$ can be expressed as $\ell(h(\boldsymbol{x}), y) = {}^{1 - y\, h(\boldsymbol{x})}\!/_2$ then $\hat{R}_n(\mathcal{H})$ can be reformulated as [6] $\hat{R}_n(\mathcal{H}) = \sup_{h \in \mathcal{H}} \left[1 - 2\hat{L}_n^{\sigma}(h)\right]$, which basically means that a space of hypothesis is small when it not able to fit random labels.

Based on the previous definitions it is possible to recall the main result of the LRC-Theory which bounds the generalization error in terms of just empirical

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

quantities. In particular, it is possible to state that with probability $(1-\delta)$ and $\forall h^* \in \mathcal{H}$

$$L(h^*) \leq \hat{L}_n^y(h^*) + \sup_{h \in \left\{h:\ h \in \mathcal{H}, \hat{L}_n^y(h) \leq \hat{L}_n^y(h^*)+\hat{c}^*\right\}} \left[1 - 2\hat{L}_n^\sigma(h)\right] + \phi_2(n,\delta), \quad (2)$$

where $\phi_2(n,\delta)$ is a confidence term and $c^* \geq 0$ is a quantity that can be computed from the data, one can refer to [5, 6] for more details. Eq. (2), with respect to Eq. (1), shows that an optimal model should not be just the result of a compromise between the accuracy of that model and the complexity of the space of models from where the model has been chosen. Eq. (2) states that, among the models with small empirical error, the functions which perform badly on random labels should be preferred.

Note also that Eqns. (1) and (2) have been successfully exploited in the past for model selection purposes, particularly on small sample problems [9, 10]. For this reason, in the next section, similarly to what has been done with Eq. (1) for SVMs, we will build a learning algorithm from Eq. (2) called Local Rademacher Complexity Machine, which tries to take advantage of the improvements of Eq. (2) over Eq. (1).

3 Local Rademacher Complexity Machine

Let us consider the same framework of SVMs [3, 4] where $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$ and $\mathcal{H}$ is defined as $\boldsymbol{w} : \boldsymbol{w} \in \mathbb{R}^d, \|\boldsymbol{w}\| \leq A \in [0,\infty)$, and $b \in \mathbb{R}$. Since we are dealing with binary classification problems the Hard loss function $\ell_H(f(\boldsymbol{x}), y) = {}^{1 - y\,\mathrm{sign}[f(\boldsymbol{x})]}/_2$ should be adopted. Unfortunately ℓ_H is not convex, then, in SVMs, the Hinge loss function $\ell_\xi(f(\boldsymbol{x}), y) = \max[0, 1 - yf(\boldsymbol{x})]$, the simplest yet effective convex upper bound of ℓ_H [15], is exploited.

Then, by following Eq. (1), and noting that $C(\mathcal{H}) \propto \|\boldsymbol{w}\|$ [3, 5], we obtain the SVMs learning algorithm

$$\min_{\boldsymbol{w},b} {}^1/_2\|\boldsymbol{w}\|^2 + C\sum_{i=1}^n \max[0, 1 - y_i f(\boldsymbol{x}_i)], \quad (3)$$

where $C \in (0,\infty)$ balances the trade-off between accuracy and complexity. Note that Problem (3) is convex and can be written in its dual form

$$\min_{\boldsymbol{\alpha}} {}^1/_2\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha}, \quad \text{s.t. } \boldsymbol{y}^T\boldsymbol{\alpha} = 0,\ \mathbf{0} \leq \boldsymbol{\alpha} \leq \boldsymbol{C} \quad (4)$$

where $\boldsymbol{\alpha} \in \mathbb{R}^n$, $\boldsymbol{v} \in \{v\}^n$ with $v \in \mathbb{R}$, $\boldsymbol{y}^T = [y_1, \cdots, y_n]$, $Q_{i,j} = y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j$, $\boldsymbol{w} = \sum_{i=1}^n \alpha_i y_i \boldsymbol{x}_i$, and b is the Lagrange multiplier of the equality constraint in Problem (4). Problem (4) also allows to exploit the kernel trick [13] and implement nonlinear models.

Eq. (2) adds a more profound intuition with respect to Eq. (1) by stating that we should not look at the complexity of the entire space $\mathcal{H}$ but we should look just to a subset of the $h \in \mathcal{H}$ with small error. Eq. (2) also gives us a way to perform this measurement: we have to check how the functions with small error behave when the labels are randomly switched. If the chosen function

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

behaves badly on random noise probably it is a function with good generalization properties.

Analogously to Eq. (3), which is based on Eq. (1), the LRCM, based on Eq. (2), can be defined

$$\min_{\boldsymbol{w},b} 1/2\|\boldsymbol{w}\|^2 + C_1 \sum_{i=1}^{n} \max[0, 1 - y_i f(\boldsymbol{x}_i)] + C_2 \sum_{i=1}^{n} \max[0, 1 + \sigma_i f(\boldsymbol{x}_i)]. \quad (5)$$

In fact, in Problem (5), we are minimizing the error over the data, measured with the Hinge loss function, and contemporary we are balancing the complexity of the solution measured with $\|\boldsymbol{w}\|^2$, but the complexity is computed by taking into account only the functions with high error over the random labels $\sigma_1, \cdots, \sigma_n$. In fact, when $C_1 \in (0, \infty)$ is small we force the solution to fit the available data with simple functions (small $\|\boldsymbol{w}\|^2$). Moreover, when $C_2 \in (0, \infty)$ is large we force also the solution to make a high error over the σ_i since $\sum_{i=1}^{n} \max[0, 1 + \sigma_i f(\boldsymbol{x}_i)] = \ell_\xi(f(x_i), \sigma_i \cdot -1)$. Note that the term $\sum_{i=1}^{n} \max[0, 1 + \sigma_i f(\boldsymbol{x}_i)]$ acts also as a random regularizer analogously to the dropout in neural networks [16]. In order to reduce the probability of unlucky realization of the σ_i with $i \in \{1, \cdots, n\}$ we will exploit the proposal described in [17] to use the Nearly Homogeneous Multi-Partitioning technique developed in [18]. The idea is to split the original dataset in two almost homogeneous subsets and assign to each of the two sets two different labels. This is an heuristic method to assign to the available samples the noisiest possible labels. In this way, the term $\sum_{i=1}^{n} \max[0, 1 + \sigma_i f(\boldsymbol{x}_i)]$, measures the capacity of the function to underfit the noisiest possible configuration of the labels.

Finally, note that Problem (5) is convex and we can compute its dual formulation

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}^T \begin{bmatrix} Q^y & Q^{y,\sigma} \\ Q^{\sigma,y} & Q^\sigma \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} - \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}, \text{ s.t. } \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{y} \\ \text{-}\boldsymbol{\sigma} \end{bmatrix} = 0, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \leq \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} \leq \begin{bmatrix} \boldsymbol{C_1} \\ \boldsymbol{C_2} \end{bmatrix} \quad (6)$$

where $Q^y_{i,j} = y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$, $Q^\sigma_{i,j} = \sigma_i \sigma_j \boldsymbol{x}_i^T \boldsymbol{x}_j$, $Q^{y,\sigma}_{i,j} = -y_i \sigma_j \boldsymbol{x}_i^T \boldsymbol{x}_j$, $Q^{\sigma,y} = (Q^{y,\sigma})^T$, $\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i + \sum_{i=1}^{n} \beta_i \sigma_i \boldsymbol{x}_i$, and b is the Lagrange multiplier of the equality constraint in Problem (6) which can be easily solved with the standard SVMs solvers [19]. Problem (6) allows, analogously to Problem (4) to exploit the kernel trick [13] and implement also nonlinear models.

4 Results and Discussion

In order to verify whether the LRCM allows to improve over the performance of the SVMs in the small–sample setting, we make use of several Human Gene Expression datasets, the same exploited in [9] (see Table 1). Since some of these datasets are multi-class problems, analogously to [9], we map the multi-class into two-class problems according to the description of Table 1.

As in this kind of setting a reference set of reasonable size is not available for evaluating the performance of the entire procedure, we reproduce the methodology suggested by [20], which consists in generating five different training/test pairs using a random sampling approach. Then the training set has

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

Id	Dataset	d	n	Class $+1$	Class -1
D01	Brain Tumor 1	5920	90	Medulloblastoma	Malignant glioma, AT/RT, Normal cerebellum, and PNET
D02	Brain Tumor 2	10367	50	Classic Glioblastomas and Anaplastic Oligodendrogliomas	Non-classic Glioblastomas and Anaplastic Oligodendrogliomas
D03	Colon Cancer 1	22283	47	Already two-class	
D04	Colon Cancer 2	2000	62	Already two-class	
D05	DLBCL	5469	77	Already two-class	
D06	Duke Breast Cancer	7129	44	Already two-class	
D07	Leukemia	7129	72	Already two-class	
D08	Leukemia 1	5327	72	ALL B-cell	ALL T-cell and AML
D09	Leukemia 2	11225	72	ALL	AML and MLL
D10	Lung Cancer	12600	203	Adeno	Normal, Squamous, COID, and SMCL
D11	Myeloma	28032	105	Already two-class	
D12	Prostate Tumor	10509	102	Already two-class	
D13	SRBCT	2308	83	EWS	RMS, BL, and NB

Table 1: Human Gene Expression datasets (see [9] for details): mapping of the multi-class into two-class problems.

Dataset	D01	D02	D03	D04	D05	D06	D07	D08	D09	D10	D11	D12	D13	# Wins
SVM	5.4 ± 0.3	**0.0** ± **0.0**	5.0 ± 0.2	4.0 ± 1.0	5.4 ± 0.3	5.0 ± 0.2	5.0 ± 0.0	**8.2** ± **0.2**	8.0 ± 0.0	14.4 ± 0.3	**6.8** ± **0.2**	6.6 ± 0.2	6.6 ± 0.2	3
$LRCM$	**4.2** ± **0.1**	**0.0** ± **0.0**	**4.1** ± **0.1**	**3.7** ± **0.2**	**4.3** ± **0.3**	**3.9** ± **0.1**	**4.1** ± **0.1**	8.7 ± 0.4	**6.8** ± **0.1**	**10.1** ± **0.3**	**6.8** ± **0.1**	**5.9** ± **0.2**	**6.2** ± **0.2**	**12**

Table 2: Human Gene Expression datasets (see Table 1): average number of errors on the test sets performed by SVM and LRCM.

been used both for training the model and for model selection purposes by exploiting the 10-fold cross validation model selection procedure. Since in our case $d \gg n$, a nonlinear formulation is generally not needed and consequently a linear SVM and LRCM is exploited. For SVM we searched for the best $C \in \{10^{-6}, 10^{-5.8}, \cdots, 10^{4}\}$. For LRCM we searched for the best C_1, C_2 in the same range.

In Table 2, we present the average number of errors, performed on the five test set replicas. In particular, we compare the results obtained with SVM and LRCM. From the results of Table 2 it is possible to note how LRCM outperforms SVM in most of the datasets in a statistical relevant way (since in the table it is reported the t-student confidence interval at 95%). Moreover, when LRCM does not outperform SVM, they both perform comparably. The only exception is the Leukemia 1 dataset, where SVM outperforms LRCM.

The present work shows for the first time that the LRC Theory, as the VC one, can be used to inspire a new learning algorithm. The presented approach is surely a preliminary tentative which needs to be extended by taking care of two main aspects. The first one is to try to check the performance of LRCM on medium and large datasets in order to check its performance on a setting which is different from the small sample one. The second problem is to test

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

the effectiveness of LRCM in non-linear problems when the kernel trick needs to be exploited. Nevertheless, these preliminary results shows the potentiality of LRCM and its effectiveness in the small-sample setting and confirm that the statistical learning theory can be an effective tool both for model selection purposes, as proved in previous work, and for inspiring new learning algorithms.

References

[1] L. Wang. *Support vector machines: theory and applications*. Springer Science & Business Media, 2005.

[2] M. Wainberg, B. Alipanahi, and B. J. Frey. Are random forests truly the best classifiers? *JMLR*, 17(1):3837–3841, 2016.

[3] V. N. Vapnik. *Statistical Learning Theory*. Wiley New York, 1998.

[4] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[5] P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *JMLR*, 3:463–482, 2002.

[6] L. Oneto, A. Ghio, S. Ridella, and D. Anguita. Global rademacher complexity bounds: From slow to fast convergence rates. *NEPL*, 43(2):567–602, 2015.

[7] P. L. Bartlett, O. Bousquet, and S. Mendelson. Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005.

[8] L. Oneto, A. Ghio, S. Ridella, and D. Anguita. Local rademacher complexity: Sharper risk bounds with and without unlabeled samples. *Neural Networks*, 65:115–125, 2015.

[9] D. Anguita, A. Ghio, L. Oneto, and S. Ridella. In-sample model selection for trimmed hinge loss support vector machine. *NEPL*, 36(3):275–283, 2012.

[10] D. Anguita, A. Ghio, L. Oneto, and S. Ridella. Learning with few bits on small-scale devices: from regularization to energy efficiency. In *ESANN*, 2014.

[11] L. Oneto, N. Navarin, M. Donini, A. Sperduti, F. Aiolli, and D. Anguita. Measuring the expressivity of graph kernels through statistical learning theory. *Neurocomputing*, 2017.

[12] C. Cortes, M. Kloft, and M. Mohri. Learning kernels using local rademacher complexity. In *NIPS*, pages 2760–2768, 2013.

[13] B. Schölkopf. The kernel trick for distances. In *NIPS*, 2001.

[14] L. Oneto, D. Anguita, and S. Ridella. A local vapnik-chervonenkis complexity. *Neural Networks*, 82:62–75, 2016.

[15] L. Rosasco, E. De Vito, A. Caponnetto, M. Piana, and A. Verri. Are loss functions all the same? *Neural Computation*, 16(5):1063–1076, 2004.

[16] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014.

[17] D. Anguita, A. Ghio, L. Oneto, and S. Ridella. Maximal discrepancy vs. rademacher complexity for error estimation. In *ESANN*, 2011.

[18] M. Aupetit. Nearly homogeneous multi-partitioning with a deterministic generator. *Neurocomputing*, 72(7):1379–1389, 2009.

[19] J. Shawe-Taylor and S. Sun. A review of optimization methodologies in support vector machines. *Neurocomputing*, 74(17):3609–3618, 2011.

[20] A. Statnikov, C.F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21(5):631, 2005.