A Tool for Training Effective Classifiers in the Small Sample Setting

Davide Anguita, Alessandro Ghio, Luca Oneto, and Sandro Ridella

DITEN, University of Genoa

Correspondence: Alessandro Ghio, Alessandro.Ghio@unige.it, +39-010-3532192, Via Opera Pia 11A, I-16145, Genoa, Italy

Abstract

We present in this work a MATLAB software tool, allowing to perform the learning phase of Support Vector Machine classifiers. Thank to the exploitation of innovative statistical methods, the models, trained and selected by the tool, result to be particularly effective in the small sample setting, i.e. when few high-dimensional data are available (e.g. in bioinformatics problems).

1 Support Vector Machine: Model Selection in the small-sample setting

The *Support Vector Machine* (*SVM*) algorithm [1], whose solid theoretical foundations date back to the development of the Statistical Learning Theory [1], represents one of the state-of-the-art techniques for classification problems and is widely used in real-world applications. The linear SVM is a classifier of the following form:

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b \tag{1}$$

where the weights $\boldsymbol{w}$ and the bias b are the parameters of the model. These parameters are computed during the *learning phase* using a *training set* $D_n \in \mathbb{R}^{n \times d}$, consisting of n d-dimensional samples. In addition to the set of parameters $(\boldsymbol{w}, b)$, the SVM algorithm requires the tuning of some *hyperparameters*, which noticeably affect the classifier performance. This tuning is not part of the training and is known as the *model selection* phase.

The problem of model selection is generally linked to the estimation of the *generalization* ability of a classifier, i.e. the ability of a model to correctly classify previously unseen data. For this purpose, the *Structural Risk Minimization* (*SRM*) principle [1] defines some basic steps for its estimation: (i) define a centroid $\hat{\boldsymbol{w}}$; (ii) choose a (possibly infinite) sequence of hypothesis spaces $\mathcal{F}_k$, $k = 1, 2, ...$, where the classes of functions $\mathcal{F}_k$ describe classifiers of growing complexity and are centered on $\hat{\boldsymbol{w}}$; (iii) select the optimal model f^o among the hypothesis spaces by exploiting the following trade-off between overfitting and underfitting:

$$f^o = \arg \min_{f \in \mathcal{F}_k, k \in 1,2,...} \mathcal{C}(f, \mathcal{F}_k) = \arg \min_{f \in \mathcal{F}_k, k \in 1,2,...} \left[\hat{L}_n(f) + \text{pen}(\mathcal{F}_k) \right]. \tag{2}$$

The first complexity term $\hat{L}_n(f)$ is the *empirical risk* or, in other words, the misclassification rate of f on D_n. It is well–known that the minimization of the empirical risk alone can lead to overfitting, if the hypothesis space is too large, but this tendency can be counterbalanced by $\text{pen}(\mathcal{F}_k)$, which penalizes complex models.

In the conventional SVM formulation, the hypothesis space is usually (and arbitrarily) centered in the origin ($\hat{\boldsymbol{w}}_0 = \boldsymbol{0}$), because, in general, there is no a–priori information leading to a better choice [2]. This choice, however, severely influences the sequence $\mathcal{F}_k$ and has a detrimental effect on the model selection of SVM classifiers. Thus, we proposed some techniques that allows to select a "good" centroid $\hat{\boldsymbol{w}}$ [2, 3] and to center a sequence of hypothesis spaces around it [2], so to better explore the classes of functions for model selection purposes. In order to estimate the complexity of the hypothesis space, captured by the penalty term, we make use of the *Rademacher Complexity* (*RC*) [4] and the *Maximal Discrepancy* (*MD*) [5] techniques, which are two statistically rigorous approaches that can be effectively used in practice [6]. These methods allow us to perform an effective model selection, especially when applied to *small sample problems*, where the number of patterns is small compared to the dimensionality of the problem [7, 8], that is where conventional out-of-sample methods, such as the K-fold Cross Validation [9], are not reliable [10, 11].

Based on these results, we realized a MATLAB routine[1] for training effective and reliable classifiers in the small sample setting, e.g. in *microarray* classification problems. The Small Sample SVM (SS-SVM) software, briefly detailed in the next Section, receives as input that the main characteristics of the experiment to be performed: the output of the software consists in the trained model, which can be used to classify new samples.

[1] The tool will be soon available for downloading at http://smartlab.ws and will be distributed under the terms of the GNU General Public Licence (GPL).

2 The SS-SVM tool

The SS-SVM tool allows to perform a complete SVM learning. As the RC and MD techniques are used for model selection purposes, the software is mainly targeted towards small sample problems, though it can also be used in other SVM training frameworks. The program SS-SVM is invoked by the command line `SSSVM (parameters)`, where the parameters contain all the information needed by SS-SVM to perform its job. What follows is a brief and simplified list of the parameters (for a complete list, please refer to the tool user guide):

- The matrix containing the training data and their labels;
- A flag, indicating whether the normalization in the $[0, 1]$ range must be applied;
- The range for the hyperparameter(s), exploited by the method. In particular, a *grid search* approach is used to exhaustively explore the hyperparameter(s) space [9];
- The method for selecting the centroids of the class of functions. The user can select between two techniques: using further unlabelled examples [3] or exploiting the algorithm proposed in [2];
- The method to be used for model selection purposes (MD or RC). Note that, depending on the chosen method, further parameters should have to be set;
- The number of Monte Carlo replicates to estimate MD or RC (e.g. refer to [2] for further details).

The output of the tool consists in the SVM model, which can be used to classify new samples (e.g. test data): the function `SSSVM_feed`, which implements the classifier of Eq. (1), can be invoked for this purpose.

3 Experimental Results

We propose some results, obtained by exploiting the SS-SVM software on Human Gene Expression (HGE) datasets: due to space constraints, we do not report here the complete list of references for the datasets, which can be retrieved in [2]. Our objective is to compare the performance of the model, selected and trained through SS-SVM, against the classifier obtained with the K-fold Cross Validation (KCV) method [9], which is considered one of the state-of-the-art approach to the SVM model selection problem: for this purpose, we reproduce the methodology used by [13], which consists in randomly generating 5 different training/test pairs using the available data. The experimental setup is the following:

- the data are normalized in the range $[0, 1]$;
- the approach proposed in [2] is exploited for selecting the centroids;
- the remaining samples are used for model selection purposes through the MD and RC approaches;
- the error rates of the optimal models, chosen by the different approaches, are then computed on the test set.

Table 1 presents both the characteristics of the dataset (where d is the number of variables and n is the number of data for training the classifier) and the average number of errors, performed by the models on the 5 randomly sampled test sets. The results clearly show that the classifiers, selected with SS-SVM (in particular, when the Rademacher Complexity technique is used), are effective in most of the cases (9 datasets out of 13). Note that a more thorough analysis will be proposed in a forthcoming paper [12], where further results are obtained comparing numerous model selection approaches on several datasets, also in the very small-sample setting ($n = 10$).

4 Conclusions

We presented in this work a MATLAB software tool for the Support Vector Machine classifiers learning. In particular, for model selection purposes, Rademacher Complexity and Maximal Discrepancy based techniques are exploited, making the tool particularly appealing with respect to conventional approaches when the user targets small sample problems.

Table 1: Average number of errors on the test sets of the HGE datasets.

Dataset	d	n	KCV	SS-SVM MD	SS-SVM RC
Brain 1	5920	90	5.5 ± 0.5	$\mathbf{5.4 \pm 0.3}$	5.6 ± 0.3
Brain 2	10367	50	2.8 ± 0.5	$\mathbf{0.0 \pm 0.0}$	0.4 ± 0.6
Colon 1	22283	47	8.7 ± 0.7	5.4 ± 0.3	$\mathbf{5.0 \pm 0.0}$
Colon 2	2000	62	8.7 ± 0.0	6.0 ± 0.0	$\mathbf{4.0 \pm 1.0}$
DLBCL	5469	77	7.6 ± 0.3	$\mathbf{5.4 \pm 0.3}$	$\mathbf{5.4 \pm 0.3}$
Breast	7129	44	6.6 ± 1.2	$\mathbf{5.0 \pm 0.2}$	$\mathbf{5.0 \pm 0.2}$
Leukemia	7129	72	$\mathbf{5.0 \pm 0.0}$	$\mathbf{5.0 \pm 0.0}$	$\mathbf{5.0 \pm 0.0}$
Leukemia 1	5327	72	9.8 ± 0.2	9.0 ± 0.0	$\mathbf{8.2 \pm 0.2}$
Leukemia 2	11225	72	9.0 ± 0.7	$\mathbf{8.0 \pm 0.0}$	$\mathbf{8.0 \pm 0.0}$
Lung	12600	203	$\mathbf{7.2 \pm 0.1}$	14.4 ± 0.3	14.4 ± 0.3
Myeloma	28032	105	$\mathbf{0.0 \pm 0.0}$	6.8 ± 0.2	6.8 ± 0.2
Prostate	10509	102	10.9 ± 1.7	6.8 ± 0.2	$\mathbf{6.6 \pm 0.2}$
SRBCT	2308	83	7.6 ± 0.5	6.8 ± 0.2	$\mathbf{6.6 \pm 0.2}$
# best			**3**	**6**	**9**

References

[1] V.N. Vapnik, "The nature of statistical learning theory", *Springer Verlag*, 2000.

[2] D. Anguita, A. Ghio, L. Oneto, S. Ridella, "Selecting the Hypothesis Space for Improving the Generalization Ability of Support Vector Machines", *Proc. of the IEEE Int. Joint Conference on Neural Networks (IJCNN)*, pp. 1169–1176, S. Jose, USA, 2011.

[3] D. Anguita, A. Ghio, L. Oneto, S. Ridella, "The Impact of Unlabeled Patterns in Rademacher Complexity Theory for Kernel Classifiers", *Advances in Neural Processing System (NIPS)*, Granada, Spain, 2011.

[4] P.L. Bartlett, S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results", *Computational Learning Theory*, pp. 224–240, 2001.

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

[6] D. Anguita, A. Ghio, S. Ridella, "Maximal Discrepancy for Support Vector Machines", *Neurocomputing*, vol. 74, pp. 1436–1443, 2011.

[7] M. Girolami, H. Mischak, R. Krebs, "Analysis of complex, multidimensional datasets", *Drug Discovery Today: Technologies*, vol. 3, pp. 13–19, 2006.

[8] A. Statnikov, I. Tsamardinos, Y. Dosbayev, C.F. Aliferis, "GEMS: A System for Automated Cancer Diagnosis and Biomarker Discovery from Microarray Gene Expression Data", *International Journal of Medical Informatics*, vol. 74, pp. 491–503, 2005.

[9] C.W. Hsu, C.C. Chang, C.J. Lin, "A practical guide to support vector classification", *Technical report*, 2003.

[10] U.M. Braga–Neto, E.R. Dougherty, "Is cross-validation valid for small-sample microarray classification?", *Bioinformatics*, vol. 20, pp. 374–380, 2004.

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

[12] D. Anguita, A. Ghio, L. Oneto, S. Ridella, "In-sample and Out-of-sample Model Selection and Error Estimation for Support Vector Machines", *IEEE Transactions on Neural Networks and Learning Systems*, in press, 2012.

[13] A. Statnikov, C.F. Aliferis, I. Tsamardinos, D. Hardin, S. Levy. "A Comprehensive Evaluation of Multicategory Classification Methods for Microarray Gene Expression Cancer Diagnosis", *Bioinformatics*, vol. 21, pp. 631–643, 2005.