ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

Using SVMs with randomised feature spaces: an extreme learning approach

Benoît Frénay and Michel Verleysen

Université catholique de Louvain - Machine Learning Group
EPL/ELEC/DICE - Place du Levant, 3 1348 Louvain-la-Neuve, Belgium

Abstract. Extreme learning machines are fast models which almost compare to standard SVMs in terms of accuracy, but are much faster. However, they optimise a sum of squared errors whereas SVMs are maximum-margin classifiers. This paper proposes to merge both approaches by defining a new kernel. This kernel is computed by the first layer of an extreme learning machine and used to train a SVM. Experiments show that this new kernel compares to the standard RBF kernel in terms of accuracy and is faster. Indeed, experiments show that the number of neurons of the ELM behind the randomised kernel does not need to be tuned and can be set to a sufficient value without altering the accuracy significantly.

1 Introduction

In classification, support vector machines (SVMs) are state-of-the-art models which have been used to solve many problems. Their success is due to three factors. Firstly, they are maximum-margin classifiers. Secondly, the dual form of the SVM optimisation problem is quadratic and its computational complexity depends on the number of data, not on their dimensionality. Thirdly, since the optimisation of their dual form only needs the inner products of data points and not the data points themselves, kernels can easily be plugged into SVMs.

When using SVMs, three choices must be made: the kernel type, the kernel parameters and the regularisation parameter. These choices are critical for the quality of the results and are usually done using cross-validation. However, this process can be computationally intensive since many models have to be built.

This paper presents a new approach inspired by the extreme learning machines (ELMs) which allows focusing on the regularisation. It proposes to build an explicit feature space using random neurons, as in extreme learning. Then, the corresponding Gram matrix is computed and used with a standard SVM. Experiments show that this approach obtains results which compare to those obtained with RBF kernels. Only the computational time is smaller in our case, because we have no kernel parameters to tune. Indeed, it appears experimentally that the dimensionality of the randomised feature space can be set to a sufficient value, e.g. 10^3, without decreasing the classification accuracy significantly. Therefore, only the regularisation constant has to be tuned.

The remaining of this paper is organised as follows. Sections 2 and 3 quickly review the SVMs and ELMs. Section 4 presents our approach mixing these frameworks. Section 5 is concerned with the experimental comparison of our approach with standard SVMs. Eventually, Section 6 concludes this paper.

2 SVMs in a nutshell

This section quickly reviews the SVM classifiers and the RBF kernel.

2.1 Maximum-margin classifiers

In binary classification, a SVM [1] is a classifier which separates the two classes by a maximum-margin hyperplane. The margin is the distance between the hyperplane and its closest data point. The idea behind SVMs is to maximise that margin, i.e. to move the hyperplane away from the data points while keeping them separated. This optimisation problem can be stated formally as

$$\left|\begin{array}{l} \min\limits_{w,b,\xi_i} \|w\|_2^2 + C\sum_i \xi_i \\ \text{s.t.} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i \end{array}\right. \tag{1}$$

where w is the weight vector (maximising the margin is equivalent to minimising the norm $\|w\|_2^2$), ξ_i is the distance between the ith data point and the hyperplane corresponding to $y_i(w \cdot x_i + b) = 1$ and C is a regularisation constant. The regularisation is necessary because the data are usually not separable. The higher the value of C, the higher the constraint and the less likely the overfitting.

There are theoretical arguments in favor of SVMs [2] and their implementation is fast. Indeed, the dual form of Eq. 1 is a quadratic optimisation problem which only involves the inner products between data points. Hence, its computational complexity depends on the number of data points, not on their dimension.

2.2 SVMs and kernels

In practice, SVMs are often used jointly with kernels. Indeed, the dual form of Eq. 1 only needs the inner products of the data. They can be computed by any valid kernel. The joint use of SVMs and kernels opens the way for using SVMs in a wide range of domains: there exist kernels to deal with real numbers, sequences, graphs etc. However, the practitioner has to (i) choose the right kernel, (ii) tune the kernel parameters (if any) and (iii) tune the regularisation parameter. This can be done using e.g. cross-validation, but it is usually computationally intensive.

In this paper, we focus on applications where the inputs are numerical. In practice, linear, polynomial and radial basis function (RBF) kernels are often used. We will compare our approach to the RBF kernel

$$k(x, z) = \exp(-\gamma\|x - z\|_2^2) \tag{2}$$

which is very common and usually gives very good results. Its only parameter is the kernel width γ which regulates the scale at which the SVM is looking at the data points. The larger the kernel width γ, the larger the scale.

3 Basis of extreme learning

This section quickly reviews the basis and implementation of extreme learning.

3.1 Extreme learning machines

In its most standard form, a MLP is a two-layer feedforward neural network with a non-linear activation function for the hidden layer and a linear activation function for the output layer. MLPs can solve classification problems, but their training relies on the slow backpropagation algorithm which is difficult to tune.

In [3], Huang et al. propose to keep the MLP structure with a different learning algorithm. First, they scale the inputs in $[-1, 1]$ and initialise the first layer weights and biases at random, e.g. in $[-3, 3]$. The n data points are then transformed by the p neurons of the hidden layer from $x_i \in \Re^d$ to $\phi(x_i) \in \Re^p$. Here, ϕ is the sigmoid function $\phi(x_i) = 1/(1+\exp(-w \cdot x_i - b))$. The output layer being linear, the problem boils down to solving a linear system with n equations and p variables

$$\Phi w = Y \tag{3}$$

where Φ is a $n \times p$ matrix which rows are the transformed data points, w is the weight vector and Y contains the labels y_i. Usually, p is much larger than d, hence the name *extreme learning machines*. The classification of a new data point x is given by $y = \text{sign}(x \cdot w)$.

3.2 Implementations

In [3, 4], Huang et al. solve Eq. 3 using the Moore-Penrose pseudoinverse and an iterative algorithm. They show that ELMs with a large enough number of neurons obtain satisfactory results in a much shorter computation time than SVMs. The number of neurons has to be tuned using e.g. cross-validation. In [5], they applied successfully the same approach to RBFs. Moreover, Miche et al. [6] and Liu et al. [7] have shown that ELMs can be pruned using LASSO or regularised w.r.t. the L2 norm of the output weights vector.

4 Proposed approach

This section describes the proposed approach merging both the SVMs and ELMs. Indeed, the transformation performed by the first layer of an ELM can be seen as a kernel which can be plugged into a SVM. The subsequent section will show that the size of the ELM is not critical and can be set to a sufficient, large value.

4.1 The ELM kernel

ELMs are fast, but they do not search for maximum-margin hyperplanes. Instead, they minimise a sum of squared errors between the class labels and the MLP output. This kind of criterion is not really suitable for classification. In this paper, we propose to merge both the SVM and ELM approaches in order to obtain models which (i) are fast to train and (ii) are maximum-margin classifiers.

In the ELM framework, we can think of the hidden layer as a mapping ϕ from the data space $\Re^d$ to a feature space $\Re^p$ where we have to solve a linear problem. This is very similar to the idea behind the use of kernels: statistically, the data points are easier to separate in a higher dimensional space. Hence, the first layer of an ELM can be thought as defining some kind of randomised kernel, which will be subsequently called the *ELM kernel.*

Given two data points x and z and an ELM with p neurons defining a mapping ϕ from $\Re^d$ to $\Re^p$, the corresponding ELM kernel function is defined as

$$k(x, z) = \frac{1}{p}\phi(x) \cdot \phi(z). \tag{4}$$

Therefore, the Gram matrix corresponding to this ELM kernel and a set of data points can be computed as

$$G = \frac{1}{p}\Phi\Phi'. \tag{5}$$

In [7], Liu et al. also propose to use the explicit mapping described above. However, their extreme SVMs use neither the maximum-margin principle nor the advantages of the dual form. Indeed, they are rather doing a Ridge regression in the ELM feature space which has a time complexity depending on the dimensionality of the data. To our knowledge, the ELM kernel has never been used jointly with real SVMs which (i) search for maximum-margin hyperplanes and (ii) offer a time complexity which is particularly interesting when using an ELM kernel computed in a high dimensional feature space.

4.2 Merging extreme learning and SVMs

Since Huang et al. showed that the ELM kernel gives good results if jointly used with a linear classifier, it makes sense to question whether the same holds for SVMs. Therefore, in the rest of this paper, we will assess the results obtained by plugging the ELM kernel into a standard SVM. At this point, we still have to tune the number of hidden neurons. In fact, we experimentally show in Section 5 that this parameter is not critical. It can be set to a large value without significantly altering the classification results, thanks to the regularisation.

5 Experiments

In this section, we study the evolution of the classification results and show that it is sufficient to set the dimensionality p of the randomised feature space to a large value p_s, e.g. 10^3.

The seven datasets used here are coming from the UCI machine learning repository: Bupa liver disorders (Bupa), Wisconsin diagnostic breast cancer (Cancer), Pima indians diabetes (Diabetes), spectf heart (Heart), Johns Hopkins University ionosphere dataset (Ion), Parkinsons disease (Parkinsons, see [8]) and sonar mines vs rocks (Sonar, see [9]).

For each dataset, we used a double 10-fold cross-validation. Each dataset is split into 10 folds which are alternatively used as test sets for the models built on the 9 remaining folds. These training data are then in turn split into 10 folds which are alternatively used as validation sets for the models built on the 9 remaining folds.

For SVMs using the RBF kernel, C values are $10^{-1:.2:3}$ and γ values are $10.^{-5:.2:2}$. For SVMs using the ELM kernel, C values are $10^{-1:.2:4}$ and p values are $\lceil 10^{0:.2:4} \rceil$. The SVMs have been trained using the LIBSVM library [10].

5.1 Behaviour during optimisation

Fig. 1(a) and Fig. 1(b) show the evolution of the test accuracy obtained by SVM classifiers using RBF and ELM kernels as the kernel width γ and the dimensionality p increase. Using RBF kernels, the accuracy increases, reaches its maximum and then decreases. In contrast, the accuracy with ELM kernels quickly stabilises for each dataset.

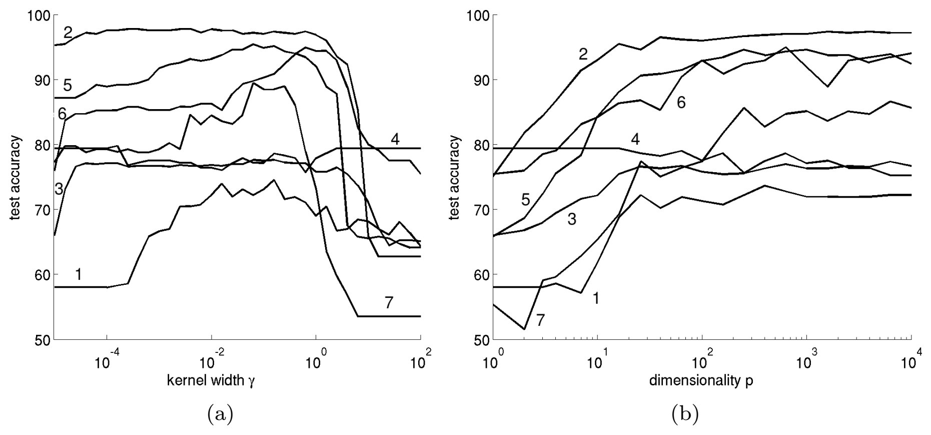

Figure 1: Test accuracy for (a) RBF and (b) ELM kernels, with respect to dimensionality p. The curve labels are given in Tab. 1.

5.2 Classification results

Tab. 1 shows the results obtained with standard SVMs using RBF and ELM kernels, i.e. the test accuracy in % and the computational time in seconds with 95% confidence intervals into parentheses. Each result is provided for the sufficient dimensionality $p_s = 10^3$ and for the optimal dimensionality p^*, which is different for each problem and selected using cross-validation.

The results show that the three models obtain very similar results, except for the training time which is much smaller for our approach when using $p = p_s$. This difference is due to the absence of additional parameters to tune. Moreover, we can see that the choice of the dimensionality p is not significantly critical for the classification performances. In fact, this is due to the regularisation which prevents the SVM from overfitting the data. This regularisation results in a model that efficiently benefits from the high-dimensional information from the ELM kernel.

		RBF kernel		ELM kernel (p^*)	ELM kernel (p_s)	
		Acc.	Comp. time	Acc.	Acc.	Comp. time
1	Bupa	72 (69−76)	140 (139−141)	71 (64−77)	72 (64−80)	5 (5−5)
2	Cancer	97 (95−98)	292 (290−293)	97 (96−98)	97 (96−98)	3 (3−3)
3	Diabetes	76 (73−80)	594 (586−602)	76 (74−78)	76 (74−79)	27 (27−28)
4	Heart	77 (72−82)	100 (98−101)	75(70−80)	77 (72−83)	1 (1−1)
5	Ion	95 (92−98)	169 (168−170)	95 (92−97)	95 (92−97)	1 (1−2)
6	Parkinsons	93 (91−96)	36 (35−36)	92 (88−96)	92 (88−96)	1 (1−1)
7	Sonar	89(83−95)	96 (95−96)	87 (83−90)	85 (83−88)	1 (1−1)

Table 1: Test accuracies and computational times for RBF and ELM kernels.

6 Conclusion

This paper proposes an approach merging both the SVM and ELM framework. Using a kernel defined using the first layer of an ELM, experiments show that the accuracy of SVM classifiers compares to the accuracy obtained with standard RBF kernels. Moreover, it appears that the only parameter of this ELM kernel, i.e. the number of neurons of the ELM, can be set to a large value without altering significantly this accuracy. As a consequence, the computational time is reduced since there are no kernel parameters left to tune.

Possible directions for further work include testing the ELM kernel on more datasets (e.g. biological datasets) and using the ELM kernel for regression.

References

[1] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[2] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* The MIT Press, 2001.

[3] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[4] G.-B. Huang, L. Chen, and C.-K. Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, 17(4):879–892, 2006.

[5] G.-B. Huang and C.-K. Siew. Extreme learning machine with randomly assigned RBF kernels. *International Journal of Information Technology*, 11(1):16–24, 2005.

[6] Y. Miche, A. Sorjamaa, and A. Lendasse. OP-ELM: Theory, experiments and a toolbox. In *ICANN*, volume 5163 of *Lecture Notes in Computer Science*, pages 145–154. Springer, 2008.

[7] Q. Liu, Q. He, and Z. Shi. Extreme support vector machine classifier. In *PAKDD*, pages 222–233, 2008.

[8] M. A. Little, P. E. McSharry, E. J. Hunter, and L. O. Ramig. Suitability of dysphonia measurements for telemonitoring of parkinson's disease. *IEEE Transactions on Biomedical Engineering*, 56(4):1015–1022, 2009.

[9] R. P. Gorman and T. J. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1:75–89, 1988.

[10] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.