SVM CLASSIFIERS WITH MODERATED OUTPUTS FOR AUTOMATIC CLASSIFICATION IN MOLECULAR BIOLOGY

A. Madevska-Bogdanova[1], D.Nikolik[2]

[1]Institute of Informatics, Faculty of Natural Sciences and Mathematics
Sts. Cyril and Methodius University
Arhimedova bb, P.O.Box 162, Skopje, Macedonia
[2]Maastricht Shnool of Management
Endepolsdomein, 150, 6229 EP Maastricht, The Netherlands
ana@ii.edu.mk, nikolik@msm.nl

Abstract: We present an alternative way of interpreting and modifying the outputs of the Support Vector Machine (SVM) classifiers – method MSVMO (Modified SVM Outputs). Stemming from the geometrical interpretation of the SVM outputs as a distance of individual patterns from the hyperplane, allows us to calculate its posterior probability i.e. to construct a probability-based measure of belonging to one of the classes, depending on the vector's relative distance from the hyperplane.

We illustrate the results by providing suitable analysis of three classification problems and comparing them with an already published method for modifying SVM outputs.

Keywords: Support Vector Machines; pattern classification; modified outputs; post-processing; posterior probability

1. Introduction

The idea behind the SVM method is rather simple: the N–dimensional vector **x** of the input problem space X is mapped with an appropriate kernel function to higher-dimensional vector of the feature space where the pattern discrimination is simpler, i.e. where there exits a linear separating hyperplane.

Let $\mathbf{x}_i$, i=1,..,N be a set of data points and o_i, i=1,.., N values indicating the corresponding classes. The goal of the support vector learning is to obtain a classifier of the form

$$\mathrm{sign}\left(\sum_{i=1}^{N_{sv}} \alpha_i o_i k(\mathbf{x}, \mathbf{x_i})\right) + b \qquad (1)$$

to determine the class of the unseen vector **x.**

N_{sv} is a number of Support Vectors (SV) and $k(\mathbf{x}_i,\mathbf{x}_j)$ is a kernel function, in different forms:

- linear SVM: $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$;
- polynomial SVM: $k(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^d$;
- Gaussian radial basis function SVM: $k(\mathbf{x}_i,\mathbf{x}_j) = \exp(-g \,\|\, \mathbf{x}_i-\mathbf{x}_j \,\|^2)$;

or other functions that satisfy the Mercer's condition (Vapnik, 1995).

As classifiers, SVM works in two phases:

phase 1: training – building a model;

phase 2: classification.

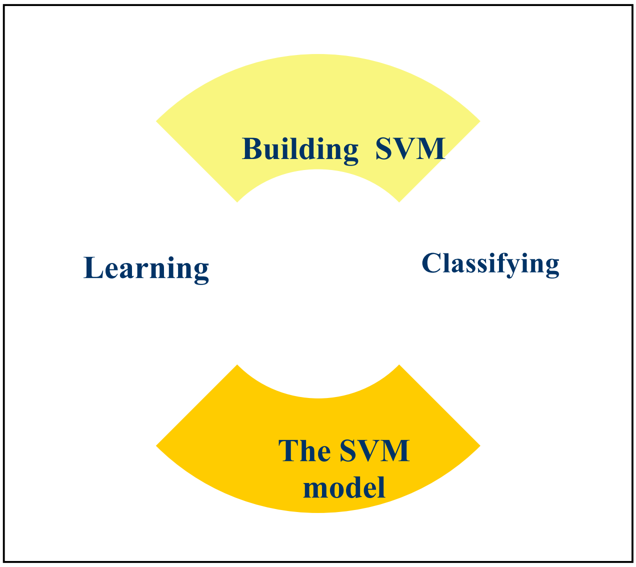

Figure 1: Schematic representation of a SVM classifier

2. The MSVMO method

The following form is the *MSVMO posterior probability* (Madevska, 2000)*:*

$$P(C \setminus \mathbf{x}) = \frac{1}{1 + \exp(k \cdot d(\mathbf{x}))} = \frac{1}{1 + \exp\left(k \cdot \frac{f(\mathbf{x})}{\|\mathbf{w}\|}\right)}, \qquad (2)$$

where f(**x**) is the SVM output for the input vector **x**, and

$$\pm\,d(\mathbf{x}) = \frac{\mathbf{x}^T\mathbf{w} + b}{\|\mathbf{w}\|} = \frac{f(\mathbf{x})}{\|\mathbf{w}\|} \qquad (3)$$

is the distance of the input vector **x** from the estimated decision hyperplane with the corresponding norm $\|\mathbf{w}\|$.

The corresponding probability for the second class is simply given by

$$P(C_2 \setminus \mathbf{x}) = 1 - P(C_1 \setminus \mathbf{x}). \qquad (4)$$

The monotonicity of (2) is assured for k < 0.

One can notice that the sign of the SVM output f(**x**) is integrated in the calculation of the probability P(C \ **x**). The negative sign indicates that the test vector **x** belongs in the class labeled with '-1' and the appropriate posterior is 1-P(C \ **x**).

The results we have obtained using the MSVMO model will never be worse than the results from the hard SVM classification.

3. Discussion

The output of the SVM algorithm produces a hyperplane, essentially a linear discriminant of the form

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b. \qquad (5)$$

In order to ensure that the logistic sigmoid function used over this linear discriminant provides a posterior probability, we have to consider the form of the discriminant itself.

This form of linear discriminant is appropriate to use provided the two-class classification problem has input vectors whose class-conditional densities are given by any exponential family of distributions with tied scale parameters (such as Gaussian, binomial, Bernoulli, Poisson). Following the Bayes theorem, the posterior probability of membership of class C_1 is therefore given by (Bishop, 1995):

$$P(C_1 \setminus \mathbf{x}) = \frac{p(\mathbf{x} \setminus C_1)P(C_1)}{p(\mathbf{x} \setminus C_1)P(C_1) + p(\mathbf{x} \setminus C_2)P(C_2)} = \frac{1}{1 + \exp(-a)} = g(a) \qquad (6)$$

where g(a) is the logistic sigmoid function given by:

$$g(a) = \frac{1}{1 + \exp(-a)}, \qquad (7)$$

and

$$a = \mathbf{w}^T\mathbf{x} + b, \qquad (8)$$

which is the same expression as (5).

We can assume that the class-conditional densities are expressed by

$$p(\mathbf{x} \setminus C_1) = \exp\{A(\Theta_1) + B(\mathbf{x},\varphi) + \Theta_1^T \mathbf{x}\} \qquad (9)$$

The form (9) represents the exponential family of distributions. The parameters Θ_1 and φ control the form of the distribution. Applying Bayes theorem, the posterior probability for class C_1 we obtain the expressions (7) and (8), where the appropriate values for $\mathbf{w}$ and b are given by:

$$\mathbf{w} = \Theta_1 - \Theta_2, \qquad b = A(\Theta_1) - A(\Theta_2) + \ln\frac{P(C_1)}{P(C_2)} \qquad (10)$$

4. Simulation results

We have used linear kernels as well as Gaussian and polynomial kernels. In the linear kernel case, the presumption for existence of the exponential class-conditional density of the data in the feature space is satisfied and the use of our logistic model for obtaining the posterior probability is hence justified.

If the learning process does not converge when linear kernels are used, we can always use non-linear ones. The use of a sigmoidal non-linearity does not require that the underlying distributions should be from the exponential family. Indeed, logistic regression has been widely used with great success by statisticians for many decades for a wide variety of applications.

The results we have obtained using linear kernels, are comparable to the ones using non-linear kernels and in some cases, they are even better (concerning the SVM classificator results.)

For illustration purpose of the method, we present the simulation results on a toy and two real problems in the molecular biology domain. We are comparing the results from a ‘hard’ classification – the outputs from the SVM classifier (1) and Platt’s modified outputs (Platt, 1999) with the results from our modified outputs – MSVMO (2). The main difference in the last two approaches is that the Platt’s modification considers the vector’s SVM output, whereas the MSVMO treats its relative distance from the hyperplane.

First, we consider the linear case, i.e. when the data are linearly separable. There are two classes, represented with circles-class ‘1’ and crosses-class ‘–1’ . As shown, the SVM algorithm correctly chooses the closest points to the separating hyperplane as its Support Vectors (the shapes in bold). The results are similar for all three cases:

- **SVM outputs**: the signs are correct, no misclassification;
- **Platt’s outputs**: correctly classified by calculating the probability;

- **MSVMO**: the calculated probabilities of the outputs, indicates that all vectors are 'on the safe distance' from the separating hyperplane.

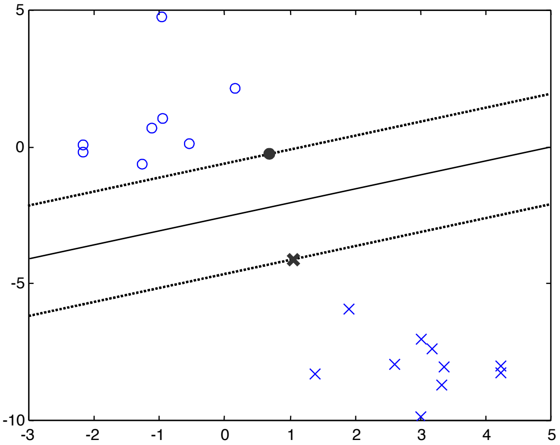

Figure 2: Linearly separable classification problem

4.1 Regulatory sequences in Arabidopsis Thaliana

Arabidopsis Thaliana has shortest genome among plants. Its regulatory sequences (genetics switches) are short (5 – 10 base pairs). Each gene has a specific pattern of regulatory sequences. They are placed 40 – 1000 base pairs upstream of a gene.

The trainings set are specifized for 1 specific element, G-box, fitting the consensus in the middle of the 50 base pairs-long sequence.

test set: 44 positive (sequences containing G-box); 4732 negative

Table 1 shows the results from the experiment. The percentages of correctly recognized vectors are given for positive and negative data separately.

4.2 Automatic classification of mitochondrial sequences

The part of the problem of protein subcellular localization is the automatic classification of mitochondrial sequences. The dataset is consisted only with human mitochondrial genes. The coding is done over the nucleotides from the maturated DNA.

test set: 233 positive,126 negative

Table 2 shows the results from the experiment with mitochondrial sequences. The percentages of correctly recognized elements are given for positive and negative data separately.

kernel	data	SVM	Platt's modified outputs	MSVMO (k=-4.84)
linear	pos	66%	77%	66%
	neg	99%	98%	99%

GRBF
C=500
G = 0.005
	data	SVM	Platt's modified outputs	MSVMO (k=-47.58)

	pos	64%	77%	64%
	neg	99%	98%	99%

Table 1: Results from Arabidopsis Thaliana regulatory sequences

kernel	data	SVM	Platt's modified outputs	MSVMO (k=-3.42)
linear	pos	71.3%	71.2%	71.3%
	neg	54.8%	54.8%	54.8%
	total	65.5%	65%	65.5%

GRBF
C = 500
g= 0.0005
	Data	SVM	Platt's modified outputs	MSVMO(k=-123.5)

	pos	73%	59.65%	73%
	neg	60%	65.1%	60%
	total	68%	61%	68%

Table 2: Percentage of correctly recognized mitochondrial sequences

5. Conclusion

We have presented an alternative way of modifying the outputs of the SVM classifiers so that a probability interpretation could easily be achieved. It is very important to assign a suitable 'measure of belonging' to a vector of a given class, which later can allow post-processing of the data set. The analytical geometry interpretation has been used as a valuable tool to link the results to the posterior probability. The outputs of the MSVMO method provide different possibility for post-processing SVM outputs. Future work will concentrate on the explanation of noticed independence of the posterior probability values regardless of the SVM classification model and differences in the results when diverse kernel functions are used.

6. References

1. Baldi P., Brunak S., Bioinformatics, the Machine Learning Approach, MIT Press, 1998
2. Bishop C. , Neural Networks for Pattern Recognition, Oxford Press, 1998
3. Burges K.J.C. "A tutorial on Support Vector Machines for Pattern Recognition", Data mining and knowledge discovery, 1998
4. Cristianini N., Shawe-Taylor J., An introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press, 2000
5. Hua S. and Sun Z., "Support vector machine approach for protein subcellular localization prediction", Bioinformatics 8, pp. 721-728, 2001
6. Joachims T., "Making large-scale SVM learning practical", in: Advances in Kernel Methods – Support Vector Learning, ed. by B. Scholkopf, K.J.C. Burges and A.J. Smola, MIT Press, pp. 169-184, 1999
7. Madevska-Bogdanova A., Nikolic D., "Automatic Classification With Support Vector Machines In Molecular Biology", Proceedings of III International Conference on Cognitive and Neural Systems, Boston, MA, USA, 1999
8. Madevska-Bogdanova A., Nikolic D., "A new approach of modifying SVM outputs", Proceedings of IJCNN'2000, IEEE-INNS-ENNS International Joint Conference on Neural Networks, Como, Italy, 2000
9. Platt J.C., "Probabilistic Outputs for SVM and comparison to Regularized Likelihood Methods", in:*Advances in Large Margin Classifiers*, ed. by A. Smola, P. Bartlett, B. Scholkopf, D. Schuurmans, MIT Press,1999
10. Richard M.D.and Lipmann R.P., "Neural network classifiers estimate Bayesian a-posteriori probabilities", Neural Computation 3, vol. 4, pp. 461-483, 1991
11. Steinwart I., "On the influence of the Kernel on the Consistency of Support Vector Machines", JMLR, pp. 267-93, 2001
12. Thomas G.B., Finney R.L.,Calculus and Analytic Geometry, Vol. 2, Addison Wesley Longman, Inc.,1998
13. Vapnik V., The nature of statistical learning theory , Springer,1995