Recognition of Human Mitochondrial Sequences Using SVM

Ana Madevska-Bogdanova[1], Dragan Nikolik[2] and Leopold Curfs[3]

[1]Faculty of Natural Sciences and Mathematics, 1000 Skopje, FYRO Macedonia
E-mail: ana@pmf.ukim.edu.mk
[2]Maastricht School of Management, Maastricht, The Netherlands
E-mail: nikolik@msm.nl
[3]Department of Genetics, Academic Hospital/University of Maastricht
Maastricht, The Netherlands
E-mail: curfs@msm.nl

Abstract. Support Vector Machines (SVM) classifiers are applied to problem in Molecular Biology - recognizing mitochondrial sequences in the human genome. We present the results obtained by SVM hard classification, using the Plat's model and Modified SVM outputs (MSVMO) method, an alternative way of interpreting and modifying the outputs of the SVM classifiers.

1 Introduction

Recognition of the human mitochondrial sequences is part of the protein subcellular localization problem, which is a key functional characteristic of proteins. A fully automatic prediction system is required, especially for the analysis of large-scale genome sequences. Experimental determination of subcellular location is mainly accomplished by three approaches: cell fractionation, electron microscopy and fluorescence microscopy. As currently practiced, these approaches are time consuming, subjective and variable. The assignment of the function for a given protein has proved to be especially difficult where no clear homology to proteins of known function exists [Bork, 1994].

Proteins need to be sorted to one or the other subcellular compartment to perform their functions. Sorting usually relies on the presence of an N-terminal targeting sequence, which is removed after entry in the right organelle. The system for automatic classification should 'learn' to recognize the mentioned targeting sequences.

Humans are part of the eukaryotic species. Mitochondrial subcellular localization is one of the four location categories: nuclear, cytoplasmic, mitochondrial and extracellular.

2 SVM and MSVMO

The support vector machines [Vapnik, 1995] is a new tool for prediction and function estimation.

Given a training set of input-output pairs

$$(\mathbf{x_1},y_1)\ ,(\mathbf{x_2},\ y_2)\ ...\ (\mathbf{x_n},y_n),$$

the **SVM** algorithm estimates a function f such that, for ($\mathbf{x}$,y) drawn according to the same distribution P($\mathbf{X}$,Y) as the training set, f($\mathbf{x}$) = y.

The **SVM** presented here is an extension of the perceptron algorithm. The perceptron learns a linear discriminant function

$$g(\mathbf{x}) = \mathrm{sign}(\mathbf{w} \cdot \mathbf{x}_i + b) \qquad (1)$$

The **SVM** extend this algorithm in two respects. It introduces non-linear decision surfaces, and a means of avoiding overfitting. The latter will be explained below. The former is achieved through a non-linear projection of the data into a higher dimensional feature space prior to estimation of the linear discriminant. The linear model learned in this space *is equivalent* to a non-linear model in the input space.

The second extension of the perceptron algorithm concerns capacity control or regularization. The **SVM** achieves good generalization by choosing a discriminant function that maximally separates the two classes in the feature space. The Euclidean distance between the closest point and the decision surface is known as the *margin.*

Maximizing the margin acts as a form of regularization. The **SVM** algorithm can be formulated in such a way that it only requires the calculation of the dot product *, between training points. Moreover in test or prediction phase, test points also only occur as dot products (1).

An algorithm with this feature is known as having a `dual form'. The **SVM** algorithm exploits the dual form by finding functions that perform the non linear projection described above, and the dot product in one step. These kernel functions' K(x,y) equate ($\mathbf{x_i}$) *($\mathbf{x_j}$) . The positions of the points in the feature space are in fact never calculated. There are many choices of kernel function, some of which have implicit feature spaces of infinite dimension. Such feature spaces provide a large number of models.

Some usual forms of the Kernel functions, are:

- 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: $k(\mathbf{x}_i,\mathbf{x}_j) = \exp(-g \, \| \mathbf{x}_i - \mathbf{x}_j \|^2)$;

The support vectors are the closest points to the separating hyperplane. They are lying at the same distance from either side of the hyperplane, assigned (+) or (-) respectively. The SVM procedure establishes that no other correctly classified vector from the training set lies closer to the hyperplane than any of the chosen support vectors.

Using the analytical geometry approach we are able to provide an alternative explanation of the vector outputs. It is very important to assign a suitable 'measure of belonging' to a vector of a given class, which later can allow post-processing the data set.

As mentioned in the beginning of this section, the aim of the SVM classifier is to classify correctly given a two-class classification problem. The output in a linearly separable case has the form:

$$f(\mathbf{x}) = \mathbf{w}^{T}\mathbf{x} + b \tag{2}$$

where **x** is an input vector.

It is clear that for a given hyperplane described with the equation $\mathbf{w}^{T}\mathbf{x}$ +b=0, and for a vector **z** that does not belong to the hyperplane, the following is satisfied:

$$\mathbf{w}^{T}\mathbf{z} + b = \pm d\|\mathbf{w}\|, \tag{3}$$

where d is 'the distance' of the 'point' **z** to the given hyperplane. The different signs determine the vector's **z** side of the hyperplane.

Equation (4) represents the *MSVMO posterior probability* [Madevska, 2000]:

$$P(C \setminus \mathbf{x}) = \frac{1}{1+\exp(k \cdot d(\mathbf{x}))} = \frac{1}{1+\exp\left(k \cdot \frac{f(\mathbf{x})}{\|\mathbf{w}\|}\right)} \tag{4}$$

The monotonicity of (4) is assured for k < 0.

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

3 Automatic Classification of Mitochondrial Sequences

The dataset of mitochondrial sequences is consisted only with human mitochondrial genes. The coding is done over the nucleotides from the maturated DNA. To dedicate to the given problem, we have created new data base, different from already prepared for subcellular localization [Reinhardt and Hubbard, 1998].

We created the data set from the repository of human mitochondrial genes from the MITOP project web (link to EMBL) and the Gene Bank for the *positive examples*, and *negative examples* sent from Maastricht Genetics Department. The positive examples are the ones that are well defined - begin with the start codon ATG, and finish with the proper stop codon.

We were interested in recognizing only the mitochondrial genes, so the models are created to recognize mitochondrial vs. non-mitochondrial sequences. In the literature there are always models that recognize one of the 4 location categories subcellular sequences in eukaryotes, and mitochondrial is one of the four classes;

The datasets in the published literature are mixture of subcellular sequences from different eukaryotic organisms. Our dataset is consisted only with human mitochondrial genes. Coding of the input sequences is one of the most important issues in building models for automatic recognition. Our choice was coding the nucleotides from the maturated DNA by:

a – 1000; c – 0100; g – 0010; t – 0001

This kind of encoding has several advantages: we work in a 4 – dimensional coding space, instead of choosing to encode the equivalent protein sequence (every triplet of nucleotides correspond to one-of-20 amino acid) and work in a 20-dimensional space. Also, this encoding offers better discriminative behavior.

3.1 Comparison to the Published Results

So far, the only published material for protein subcellular localization prediction using SVM approach is from the authors [8]. They used the amino acid composition for representing the input sequences: the input vector dimension is 20, and each unit represents the percentage of each amino acid in the protein sequence. The training data set is consisted of 321 positive examples (mitochondrial sequences) and 2106 negative (nuclear, cytoplasmic an extracellular eukaryotic sequences). In the following table is given their best result. This result was much better in recognizing the other subcellular sequences.

SVM kernel	polynomial kernel (%)	RBF (%)
	46.1	**56.7**

Table 1. Percentage of correctly recognized mitochondrial sequences [Hua, Sun 2001]

In order to compare our results to the published ones by Hua and Sun, we encoded *our data base* by amino acid composition, build SVM classifier and tested it on the test part of the data set. The same data set (the same ratio of training/test data) was used to build another SVM model, when the input vectors are encoding the nucleotides from the maturated DNA.

During the preparation of this series of experiments, we noticed that there are some mitochondrial sequences that contained nucleotides that could not all be grouped in the triplets (their total number of nucleotides could not be divided by 3, so the last ones could not be translated in the proper amino acid). To avoid any incorrectness, these sequences were removed from the data base.

Total number of data:	383 positive and 273 negative
Train set:	297 vectors (150 positive and 147 negative)
Test sets:	359 vectors (233 positive and 126 negative).

- amino acid composition (Hua and Sun)

g	16	5	0,5	**0,005**	0.0005	0.00005	0.000005
%	1	1	1,39	**49,58**	34,54	33,42	32,86

Table 2. Results of different SVM GRBF models for the amino acid composition encoding **GRBF Kernel, C=500. Best result** on the test set: **49.58**%.

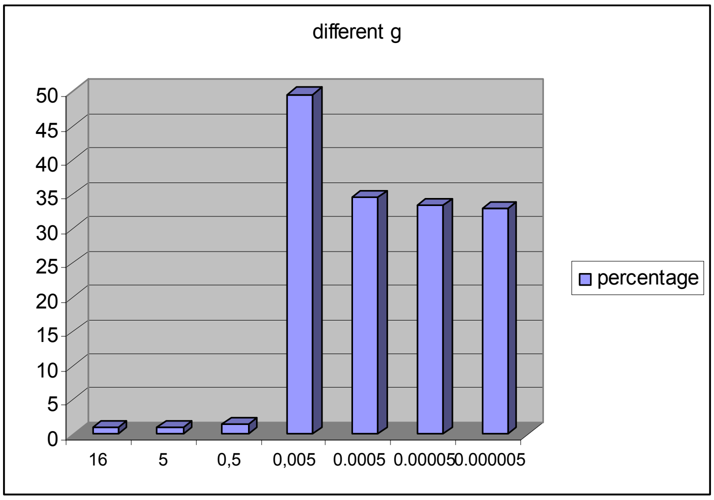

Fig. 1. Graphical interpretation of the results from Table 2

- Maturated DNA encoding

g	5	0.5	0,005	**0,0005**	0.00005	0.000005
%	FP	FP	FP	**68**	65.2	67.13

Table 3. Results of different SVM GRBF models for the maturated DNA encoding **GRBF Kernel, C=500** (FP – false positives**). Best result** on the test set: **68**%.

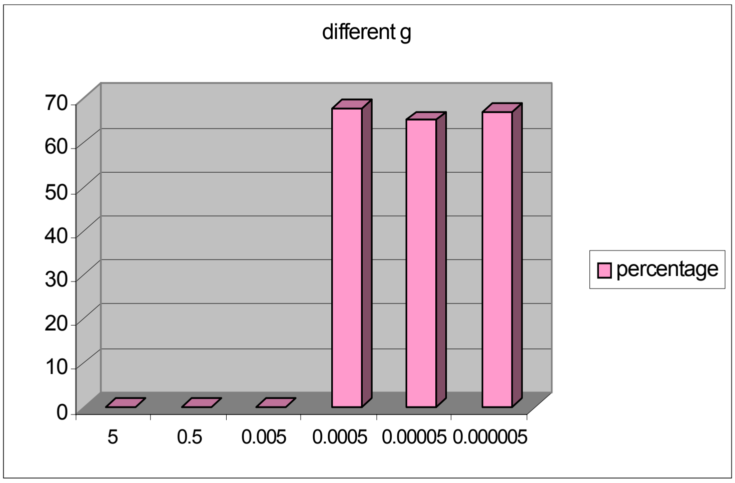

Fig. 2. Graphical interpretation of the results from Table 3

3.2 Linear and GRBF Kernel

Further experiments were taken out to point out the mining of the MSVMO algorithm over the same problem.

The linear SVM model was build and the results were compared to the results from the best choice of the GRBF Kernel, described in the previous section.

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).

We are comparing the results from a 'hard' classification – the outputs from the SVM classifier and Platt's modified outputs [9] with the results from our modified outputs – MSVMO (4).

Table 4 shows the results from the experiment with mitochondrial sequences. The percentages of correctly recognized elements are given for positive and negative data separately.

Kernel		**SVM**	**Platt**	**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**		**SVM**	**Platt**	**MSVMO** **(k=-123.5)**
	pos	73%	59.65%	73%
	neg	60%	65.1%	60%
	total	**68%**	**61%**	**68%**

Table 4. Percentage of correctly recognized mitochondrial sequences sizes

The next figure presents the comparation between the probabilities calculated for the linear and Gaussian SVM classification model.

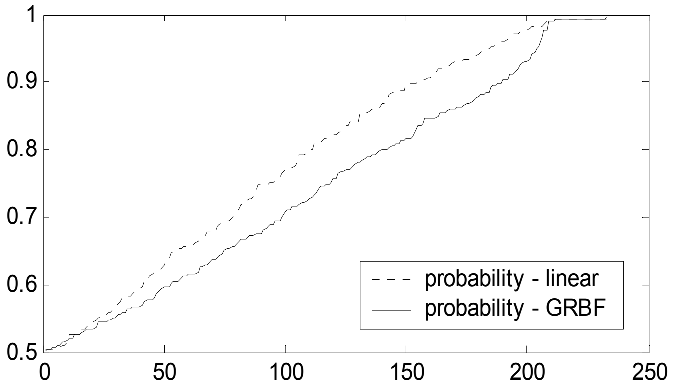

Fig. 3. Comparation between the MSVMO probabilities (abs) of the linear and Gaussian SVM model in the mitochondrial sequence recognition problem

4 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 used over a Molecular Biology problem. 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 outputs of the MSVMO method provide different possibilities for post-processing the SVM outputs.

The obtained results for the given problem: recognition of the human mitochondrial sequences – the published and the ones presented in the paper are in favor of our modeling. There are some differences in the two approaches: representation of the data and content of the data base - human mitochondrial sequences vs. sequences of different eukaryotic organisms.

References

[1] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford Press, (1998)

[2] V.N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, (1995)

[3] N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*, Cambridge University Press, (2000)

[4] G.B. Thomas, R.L. Finney, *Calculus and Analytic Geometry*, Vol.2, Addison Wesley Longman,(1998)

[5] P.E. Gill, W. Murray, M.H. Wright, *Practical Optimization,* Academic Press, (1981)

[6] P. Baldi, S. Brunak, *Bioinformatics, the Machine Learning Approach*, MIT Press, (1998)

[7] I. Steinwart, On the Influence of the Kernel on the Consistency of Support Vector Machines, *JMLR* 2:67-93, (2001)

[8] S. Hua and Z. Sun, Support Vector Machine Approach for Protein Subcellular Localization Prediction, *Bioinformatics* 8:721-728 (2001)

[9] J.C. Platt, Probabilistic Outputs for SVM and Comparison to Regularized Likelihood Methods, In: A. Smola, P. Bartlett, B. Scholkopf, D. Schuurmans (eds.) *Advances in Large Margin Classifiers*, MIT Press, (1999)

[10] G. Wahba, Support Vector Machines, Reproducing Kernel Hilbert Spaces and Randomized GACV, In: B. Scholkopf, K.J.C. Burges and A.J. Smola (eds.) *Advances in Kernel Methods – Support Vector Learning,* MIT Press, (1999) 69-88

[11] T. Joachims, Making Large-scale SVM Learning Practical, In: B. Scholkopf, K.J.C. Burges and A.J. Smola (eds.) *Advances in Kernel Methods – Support Vector Learning*, MIT Press, (1999), 169-184

[12] M.D. Richard and R.P. Lipmann, Neural Network Classifiers Estimate Bayesian a-posteriori Probabilities, *Neural Computation* 3(4):461-483, (1991)

[13] J.B. Hampshire, B. Perlmutter, Equivalence Proofs for Multilayer Perceptron Classifiers and the Bayesian Discriminant Function, In: D.S. Touretzky, J.L. Elman, T.J. Sejnowski, G.E. Hintotn (eds.) *Proceedings Connectionist Models Summer School*, San Mateo, CA, Morgan Kaufman, (1990) 159-172

[14] K.J.C. Burges A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery (1998).

[15] A. Madevska, D. Nikolic, Automatic Classification with Support Vector Machines, *Proceedings 3rd International Conference on Cognitive and Neural Systems*, Boston, MA, (1999)

[16] A. Madevska, D. Nikolic, A New Approach of Modifying SVM Outputs, *Proceedings IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN),* Como, Italy (2000)

[17] A. Madevska, D. Nikolic, Probabilistic SVM Outputs for Pattern Recognition Using Analytical Geometry: *Neurocomputing* (2003), accepted

[18] G. Thijs, Y. Moreau, S. Rombauts , B.D. Moor, P. Rouze Recognition of Gene Regulatory Sequences by Bagging of Neural Networks, *Proceedings ICANN Conference*, Toronto, Canada (1999)

[19] P. Bork, L. Holm, and C. Sander, The Immunoglobulin Fold. Structural Classification, Sequence Patterns and Common Core, *Journal Molecular Biology* 30:309-320 (1994)

[20] A. Reinhardt, and T. Hubbard, Using Neural Networks for Prediction of the Subcellular Location of Proteins. *Nucleic Acids Research* 26**:**2230-2236 (1998)