Biological applications of support vector machines

Zheng Rong Yang

Date received (in revised form): 26th September 2004

Zheng Rong Yang
is a senior lecturer in the Department of Computer Science, University of Exeter. He has been invited both to join program committees and to chair many sessions at international conferences. He has recently organised the 5th International Conference in Intelligent Data Engineering and Automated Learning (IDEAL04). His current research interests cover pattern recognition and bioinformatics.

Keywords: *support vector machines, sequence analysis, protein function annotation, protein functional site recognition*

Abstract

One of the major tasks in bioinformatics is the classification and prediction of biological data. With the rapid increase in size of the biological databanks, it is essential to use computer programs to automate the classification process. At present, the computer programs that give the best prediction performance are support vector machines (SVMs). This is because SVMs are designed to maximise the margin to separate two classes so that the trained model generalises well on unseen data. Most other computer programs implement a classifier through the minimisation of error occurred in training, which leads to poorer generalisation. Because of this, SVMs have been widely applied to many areas of bioinformatics including protein function prediction, protease functional site recognition, transcription initiation site prediction and gene expression data classification. This paper will discuss the principles of SVMs and the applications of SVMs to the analysis of biological data, mainly protein and DNA sequences.

INTRODUCTION

With the rapid increase in size of the biological databanks, understanding the data has become critical. Such an understanding could lead us to the elucidation of the secrets of life or ways to prevent certain currently non-curable diseases such as HIV. Although laboratory experiment is the most effective method for investigating the data, it is very financially and labour expensive. Computational algorithms and tools have therefore been widely used in the fields of the classification, regression and cluster analysis of biological data. The major objective in classification analysis is to train a classification model based on labelled data. The trained model is then used for classifying novel data. For instance, a classifier can be trained on a set of HIV peptides, some of which are cleaved and some not.[1] A classifier trained on this set of labelled (ie cleaved or non-cleaved) peptides can be used to classify unlabelled HIV peptides and hence to classify them as either cleaved or non-cleaved. The information can be used by pharmaceutical companies to design suitable inhibitors to fight the disease.

Classification analysis requires two descriptions of an object. One is the set of features that will be used as inputs to train the model. The other is referred to as the class membership.

Classification analysis aims to find a mapping function from the features to the class label. There have been many computational algorithms available for the classification analysis of biological data. For instance, decision trees,[2] discriminant analysis,[3] neural networks and support vector machines (SVMs).[4] The essence in classification is to minimise the probability of error in using the trained classifier. This is referred to as the structural risk and it has been shown that SVMs are able to minimise the structural risk through finding a unique hyper-plane with maximum margin to separate data from two classes.[4] Because of this, SVM classifiers provide the best generalisation ability on unseen data compared with the other classifiers.

Many applications of SVMs to biological data analysis are discussed here. The next section briefly introduces support vector machines. This is followed by a discussion of the most important step

Zheng Rong Yang,
Department of Computer Science,
Exeter University,
Prince of Wales Road,
Exeter EX4 4PS, UK

Tel: +44 (0)1392 6614045
E-mail: Z.R.Yang@exeter.ac.uk

in using SVMs for analysing protein or DNA sequences, efficient coding of biological information contained in sequences. Then the applications of SVMs to two classification problems in biology, ie modelling whole sequences and subsequences, are discussed. Finally, possible future research directions in applying SVMs to the analysis of biological data are reviewed.

SUPPORT VECTOR MACHINES

Support vector

A classification algorithm aims to find a mapping function between input features $\boldsymbol{x}$ and a class membership $t \in \{-1,1\}$:

$$y = f(\boldsymbol{x}, \boldsymbol{w})$$

where $\boldsymbol{w}$ is the parameter vector, $f(\boldsymbol{x},\boldsymbol{w})$ the mapping function and y the output. With other classification algorithms, the Euclidean distance (error) between y and t is minimised to optimise $\boldsymbol{w}$. This can lead to a biased hyper-plane for discrimination. In Figure 1, four open circles of class A and four filled circles of the class B are distributed in balance. With this data set, the true hyper-plane separating two classes of circles can be found as in Figure 1(a). Suppose a shaded circle belonging to class B is included as seen in Figure 1(b), the hyper-plane (the dashed line) will be biased because the error (distance) between the nine circles and the hyper-plane has to be minimised. Suppose a shaded circle belonging to class A is included as seen in Figure 1(c), the hyper-plane (the dashed line) will also be biased. With these biased hyper-planes, the novel data denoted by the triangles could be misclassified.

In searching for the best hyper-plane, SVMs find a set of data points that are the most difficult training points to classify. These data points are referred to as support vectors.[4] In constructing an SVM classifier, the support vectors are closest to the hyper-plane and are located on the boundaries of the margin between two classes. The advantage of using SVMs is that the hyper-plane is searched through maximising this margin. Because of this, the SVM classifier is the most robust, and hence has the best generalisation ability. In Figure 1(d), two open circles on the upper boundary and two filled circles on the lower boundary are selected as support vectors. Only the use of these four circles can form the boundaries of the maximum margin between two classes. The trained SVM classifier is a linear combination of the similarity between an input and the support vectors. The similarity between an input and the support vectors is quantified by a kernel function defined as:

Kernel function

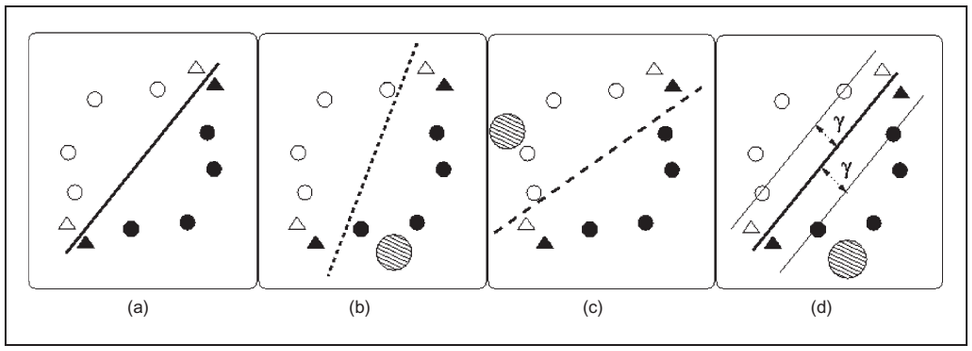

Figure 1: (a) Hyper-plane formed using conventional classification algorithms for the data with a balanced distribution. (b) and (c) Hyper-planes formed using conventional classification algorithms. (d) Hyper-plane formed using SVMs. The open circles represent class *A*, the filled circles class *B* and the shaded circle class *A* or *B*. The thick lines represent the correct hyper-plane for discrimination and the broken thick lines the biased hyper-planes. The thin lines are the margin boundaries. The triangles represent the novel patterns (see text). Gamma (γ) means the distance between hyper-plane and the boundary formed by the support vectors. The margin is 2γ

$\psi(\boldsymbol{x}, \boldsymbol{x}_i)$

where $\boldsymbol{x}_i$ is the ith support vector. The decision is made using the following equation:

Subsequence

$$\gamma = \text{sign}\{\Sigma\alpha_i t_i \psi(\boldsymbol{x}, \boldsymbol{x}_i)\}$$

where t_i is the class label of the ith support vector and α_i is the positive parameter of the ith support vector determined by an SVM algorithm. In SVMs, $\psi(\boldsymbol{x}_i)$is referred to as a feature and $\psi(\boldsymbol{x},\boldsymbol{x}_i) = \psi(\boldsymbol{x})\psi(\boldsymbol{x}_i)$. The most difficult part in SVMs is the design of a proper kernel function that corresponds to the selection of a proper number of hidden neurons in neural networks. There have been many kernel functions designed for dealing with numerical attributes. For instance, a dot product function

Feature

$$\psi(\boldsymbol{x}, \boldsymbol{x}_i) = (\boldsymbol{x}, \boldsymbol{x}_i + 1)^p$$

(p is the order of this polynomial function) or a radial basis function:

$$\psi(\boldsymbol{x}, \boldsymbol{x}_i) = \exp(-\alpha|\boldsymbol{x} - \boldsymbol{x}_i|^2)$$

(α is a constant). However, when we deal with a data set with non-numerical attributes such as protein or DNA sequences, the kernel function must be specially designed. The successful design of a proper kernel function for handling protein or DNA sequences relies on the efficient coding of the biological information contained in sequences. The next section will discuss this issue.

Coding biological information

METHODS FOR CODING BIOLOGICAL INFORMATION IN SEQUENCES

The objective of coding biological information in sequences is to provide a method for converting non-numerical attributes in sequences to numerical attributes. Before discussing this, we need to know two types of the analyses of biological sequences. The first is to analyse whole sequences aiming to annotate novel proteins or classify proteins. The second is to recognise functional sites within a sequence. The latter normally deals with subsequences which are obtained through moving a sliding window with a fixed length from the N-terminal to the C-terminal residue by residue. The residues within a scan form a subsequence. If there is a functional site within a subsequence, the subsequence is as labelled as functional, otherwise it is labelled as non-functional.

Sequence similarity

Protein annotation

Functional site recognition

There are three main methods for coding a whole sequence – the composition, profile and pairwise homology alignment methods – and two common methods for coding a subsequence – the distributed encoding and bio-basis function methods. The composition, profile and distributed encoding methods correspond to the feature extraction methods in pattern recognition. These three methods express a sequence using a vector of numerical attributes, $\boldsymbol{x}_i$. Each numerical vector is then transformed into a feature vector using a kernel function, $\psi(\boldsymbol{x}_i)$, in SVMs. While the pairwise homology alignment and bio-basis function methods do not need to extract numerical attributes, they use a specially designed kernel function, $\psi(\boldsymbol{s}_i)$, to transform each sequence, $\boldsymbol{s}_i$, directly into a feature vector in SVMs.

Composition method

With this method, we can express a protein sequence using a vector of 20 numerical attributes and a DNA sequence, a vector of 4 numerical attributes. Each numerical attribute is the occurrence of a specific amino acid or nucleic acid in the sequence. As this method does not consider any coupling effect among the neighbouring residues, other composition methods have been used, for instance, dipeptides and motif compositions and descriptors.

Profile method

With this method, each sequence is expressed as a set of similarities (probabilities) with a model or a family of sequences. The construction of a profile is normally limited to the use of positive data (functional sequences) without the

concern of discrimination. Figure 2 shows such an example using hidden Markov models (HMM).

Pairwise homology method

With the profile method, only positive sequences (sequences from one family) are used. The discrimination ability may not be satisfied. In order to enable a classifier to discriminate well, both positive and negative sequences should be used for learning. The homology alignment method for SVMs has therefore been the dominant area of research for analysing whole sequences in recent years. Rather than using nearest neighbour methods as most homology alignment tools such as BLAST[5] and its variants do, the prediction is based on a statistical model built on the relationship between sequences and support vectors selected from training sequences. Each sequence will be represented by a vector of features and each feature represents the similarity between the sequence and one of the support vector using a kernel function $\psi(\boldsymbol{s},\boldsymbol{s}_i)$, where $\boldsymbol{s}_i$ is a support vector and $\boldsymbol{s}$ a sequence. The similarity can be calculated through aligning $\boldsymbol{s}$ with the support vectors and the alignment generates a similarity between them as a feature. A collection of all the features after aligning a sequence with all the support vectors forms a feature vector. A classifier is trained using such a set of feature vectors. A feature vector will be formed in the same way for a novel sequence and is then input to the trained classifier for prediction. Figure 3 shows such an example.

Homology alignment

Nearest neighbour

Distributed encoding method

With the distributed encoding method,[6] each amino acid is encoded by a 20-bit binary vector with one bit setting as one and the rest zeros. For nucleic acids, a

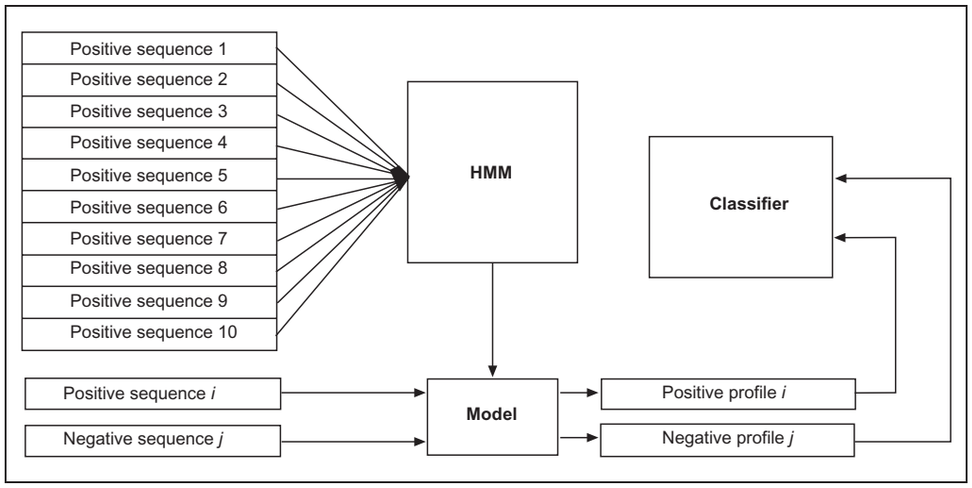

Figure 2: Ten positive sequences are used for constructing an HMM model. From this, both positive and negative sequences are input to the trained HMM model to obtain the profiles, which are used to train a classifier

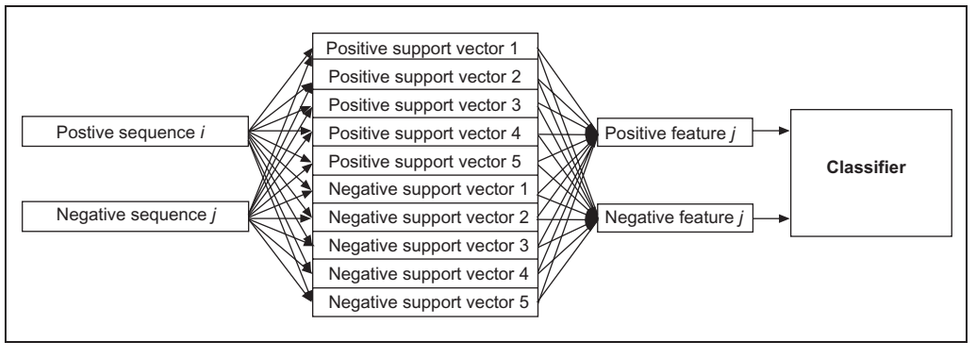

Figure 3: Pairwise homology method. Five positive and five negative support vectors are selected. Both positive and negative sequences are aligned with support vectors to produce feature vectors, which are used to train a SVM classifier

4-bit binary vector is used. In some cases, a 21-bit binary vector is used for including an 'unknown' amino acid and a 5-bit binary vector is used for including an 'unknown' nucleic acid.

Like the composition method, the distributed encoding method encounters a problem in that it is hard to code biological content in sequences. This can be seen from the fact that the distance (dissimilarity) between any pair of different amino acids or nucleic acids is always $\sqrt{2}$.[7] However, the similarity between any pair of amino acids varies (see the Dayhoff matrix in Table 1, where each entry shows a probability of mutation from one amino acid to the other).[8,9]

Dayhoff matrix

The other difficulty of the distributed encoding method is the model size. The number of input variables is enlarged 20 times for protein sequences, as shown in Figure 4.

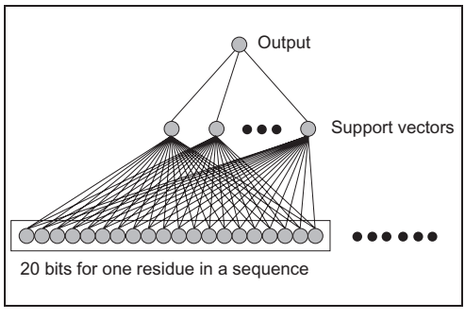

Figure 4: An example of using the distributed encoding method, where each residue is encoded using 20 inputs

Bio-basis function method

The bio-basis function was developed in 2003 for implementing the bio-basis function neural networks.[7,10] The basic principle of the bio-basis function is the normalisation of non-gapped pairwise homology alignment scores. Figure 5 shows how a query subsequence (***IPRS***) will be aligned with two support vectors (***KPRT*** and ***YKAE***) to produce two non-gapped homology alignment scores a and b respectively. Because $a > b$, it is believed that the query subsequence shares more functional similarity with the first support vector.

In summary, the composition, profile and distributed encoding methods convert sequences to numerical vectors and then

Table 1: The Dayhoff mutation matrix is based on the concept of the point-accepted mutation (PAM). An evolutionary distance of 1 PAM indicates the probability of a residue mutating during a distance in which 1 point mutation was accepted per 100 residues. For instance, the mutation probability of alanine (A) to itself is 40 and the mutation probability of alanine being substituted by cysteine (C) is 24

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	40	24	32	32	16	36	28	28	28	24	28	32	36	32	24	36	36	32	8	20
C	24	80	12	12	16	20	20	24	12	8	12	16	20	12	16	32	24	24	0	32
D	32	12	48	44	8	36	36	24	32	16	20	40	28	40	28	32	32	24	4	16
E	32	12	44	48	12	32	36	24	32	20	24	36	28	40	28	32	32	24	4	16
F	16	16	8	12	68	12	24	36	12	40	32	16	12	12	16	20	20	28	32	60
G	36	20	36	32	12	52	24	20	24	16	20	32	28	28	20	36	32	28	4	12
H	28	20	36	36	24	24	56	24	32	24	24	40	32	44	40	28	28	24	20	32
I	28	24	24	24	36	20	24	52	24	40	40	24	24	24	24	28	32	48	12	28
K	28	12	32	32	12	24	32	24	52	20	32	36	28	36	44	32	32	24	20	16
L	24	8	16	20	40	16	24	40	20	56	48	20	20	24	20	20	24	40	24	28
M	28	12	20	24	32	20	24	40	32	48	56	24	24	28	32	24	28	40	16	24
N	32	16	40	36	16	32	40	24	36	20	24	40	28	36	32	36	32	24	16	24
P	36	20	28	28	12	28	32	24	28	20	24	28	56	32	32	36	32	28	8	12
Q	32	12	40	40	12	28	44	24	36	24	28	36	32	48	36	28	28	24	12	16
R	24	16	28	28	16	20	40	24	44	20	32	32	32	36	56	32	28	24	40	16
S	36	32	32	32	20	36	28	28	32	20	24	36	36	28	32	40	36	28	24	20
T	36	24	32	32	20	32	28	32	32	24	28	32	32	28	28	36	44	32	12	20
V	32	24	24	24	28	28	24	48	24	40	40	24	28	24	24	28	32	48	82	4
W	8	0	4	4	32	4	20	12	20	24	16	16	8	12	40	24	12	8	100	32
Y	20	32	16	16	60	12	32	28	16	28	24	24	12	16	16	20	20	24	32	72

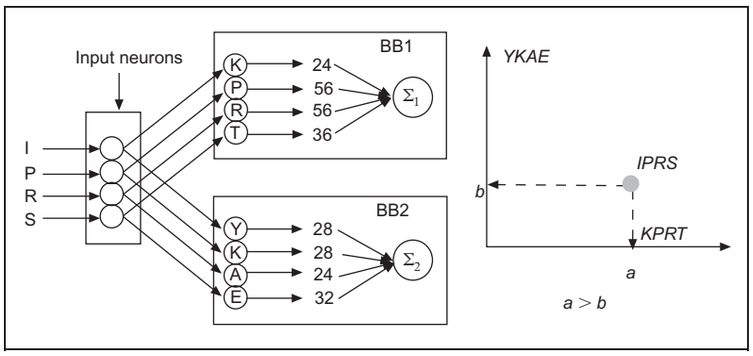

Figure 5: An illustration of the development of bio-basis (BB) function. As ***IPRS*** is more similar to ***KPRT*** than ***YKAE***, its similarity with ***KPRT*** is larger that that with ***YKAE***; see the right-hand figure

Chemical descriptor

use a kernel function to transform these numerical vectors to feature vectors. However, the pairwise homology alignment and bio-basis function methods use a kernel function to transform sequences to feature vectors directly for the use of SVMs. In terms of this, the latter is more efficient. Figure 6 gives a comparison.

APPLICATIONS

Whole sequence

The composition method has been the most popular method for analysing whole protein sequences for many years. For instance, the composition method was used for the prediction of membrane protein types,[11] the prediction of protein structural classes,[12] subcellular location prediction[13] and the prediction of secondary structures.[14] Dipeptides, gapped (up to two gaps) transitions and the occurrence of some motifs as additive numerical attributes were used to enhance the prediction of subcellular locations.[15] In the simulation it was shown that the inclusion of these additive numerical attributes did enhance the prediction accuracy. The same method has also been used in gene identification for functional RNAs in genomic sequences.[16] Instead of using transition composition to enhance the prediction performance, descriptors were also used, for instance, to predict multi-class protein folds,[17] to classify proteins[18] and to recognise rRNA-, RNA- and DNA-binding proteins.[19] SVMs also accurately discriminated cytoplasmic ribosomal protein genes from all other genes of a known function in *Saccharomyces cerevisiae*, *Escherichia coli* and *Mycobacterium tuberculosis* using codon composition, a fusion of codon usage bias and amino acid composition sign.[20]

Fisher kernel

There are two ways to generate profiles. First, a profile of a sequence can be generated by subjecting it to a homology alignment method like BLAST (Basic Local Alignment Search Tool) against a family of sequences in a database.[5] Second, a profile of a sequence can be generated using HMMs.[21,22] For instance, HMMs were used to generate profiles based on the positive sequences only and a Fisher kernel was designed for using SVMs to detect remote protein homologies.[21,22] The gradient vector of a sequence is computed with respect to the trained model. Each element of the gradient vector corresponds to a parameter of the HMMs. SVMs were trained on both positive and negative gradient vectors. Two methods (generating profiles using HMMs and homology alignment methods) have been compared for classifying G-protein coupled receptors.[23] The simulation showed that SVMs with HMM profiles performed the best. The profile method

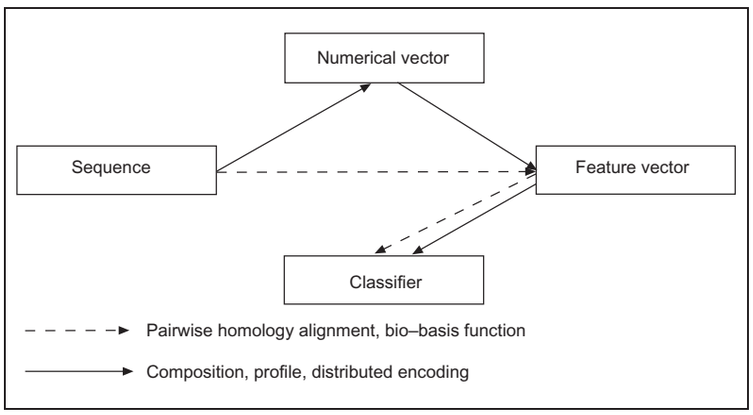

Figure 6: A comparison between two types of coding mechanisms. It can be seen that the pairwise homology alignment and bio-basis function methods directly transform sequences to features, which reduces the possibility of information loss when extracting numerical attributes

was also used for the prediction of secondary structures.[24]

Pairwise-SVM

Liao and Noble used pairwise homology alignment scores as features for training SVMs in protein homology detection.[25,26] An SVM classifier was then trained on these features. The work proved that this pairwise-SVM performed better than Fisher-SVM.[21,22]

Fisher-SVM

SVMs were also used to classify proteins with remote homology into functional and structural families based on sequence homology.[27] In that work, each feature is the occurrence of a specific *k*-mer (subsequence with *k* residues) in a sequence. Recently, SVMs were used to predict disordered regions in proteins, where a profile was formulated using PSI-BLAST (Protein Specific Iterated BLAST) for each sequence against a non-redundant sequence database.[28] Moreover, SVMs were used to detect remote homology between protein sequences, which cannot be done sufficiently when using conventional methods such as BLAST or FastA (based on the idea of identifying short 'words' or *k*-tuples common to both sequences under comparison).[29]

Bio-SVM

Subsequence

Since its invention in 1988,[6] the distributed encoding method has been widely used for the analysis of biological subsequences using SVMs. For instance, it was used for the prediction of translation initiation sites.[30] Interestingly, the work designed a novel kernel function which simply counted the number of nucleotides that coincide between two sequences. The kernel function was further improved based on the biological knowledge that local correlation information is important for translation initiation sites. It was also used for the classification of proteins with a selective kernel scaling method,[31] the prediction of the alpha and beta turns,[32] the prediction of phosphorylation sites,[33] T-cell receptor[34] and the prediction of protein–protein interactions.[35]

The bio-basis function was initially developed for implementing the bio-basis function neural network.[7] The method has been used for the prediction of trypsin cleavage sites,[7] HIV cleavage sites,[10] hepatitis C virus protease cleavage sites,[36] signal peptide cleavage sites,[37] disordered protein prediction,[38] phosphorylation site prediction[39] and the prediction of the O-linkage sites in glycoproteins.[40] In all cases, the bio-basis function neural network was more successful than other classification algorithms, such as decision trees and neural network with the back-propagation algorithm. In order to improve the performance when using the bio-basis function, bio-support vector machine (bio-SVM) was developed for the prediction of protease cleavage sites in proteins.[41] The difference between the bio-basis function neural network and bio-SVM is that the former searches for a hyper-plane that minimises the distance (error) between all the subsequences and the hyper-plane, while the latter searches for the hyper-plane that maximises the margin for generalisation. Based on this, the bio-SVM can improve the generalisation performance in analysing subsequence data.

Table 2 gives a summary of the prediction accuracy when applying SVMs to different applications as mentioned above.

FUTURE RESEARCH DIRECTIONS

Although SVMs have been widely applied to the analysis of biological sequences, some issues still need further research, particularly kernel design and negative data selection.

There are two types of kernel functions currently, residue frequency-based kernel functions and homology-based kernel functions. With a residue frequency-based kernel function, the frequency of the matched or mismatched residues from two sequences is calculated as the similarity between two sequences. For instance, a dot product kernel is a residue-matching kernel function and was used in analysing subsequences where the nucleic acids were encoded using the distributed

Table 2: A summary of the prediction accuracy of applying SVMs

Reference	ANN	HMM	BLAST	SVM
12	n.a.	n.a.	n.a.	89.20%
13	66.00%	73.00%	n.a.	79.40%
14	n.a.	n.a.	n.a.	74.00%
15	n.a.	n.a.	n.a.	72.40%
17	n.a.	n.a.	0.74*	0.78*
18	n.a.	n.a.	n.a.	86.5–99.4%
19	n.a.	n.a.	n.a.	81.00–96.8%
20	n.a.	n.a.	n.a.	87.90–97.80%
23	51.00%	70%	75.50%	86.30%
24	77.00%	n.a.	n.a.	77.30%
28	74.70%	n.a.	n.a.	75.40%
29	n.a.	n.a.	n.a.	0.87*
30	84.60%	n.a.	n.a.	88.60%
34	n.a.	n.a.	n.a.	87.90%
41	90.0%	n.a.	n.a.	91.20%
42	n.a.	n.a.	n.a.	85.40%
43	n.a.	n.a.	n.a.	75.40%

*The area under the receiver operating characteristic (ROC) curve.[44]

encoding method.[30] As the distributed encoding method produces orthogonal binary vectors, the output of this function is exactly the number of the identical nucleic acids in two subsequences. The mismatch kernel function is another residue-matching kernel function,[27] where sequence similarity was measured based on the shared occurrence of fixed-length subsequences in data. With a homology-based kernel function, the homology alignment between two sequences is calculated using a mutation matrix as the similarity between two sequences. The Fisher kernel function used in Fisher-SVMs,[21,22] the pairwise kernel function in pairwise-SVM[25,26] and the bio-basis function in the bio-SVM[41] are homology kernel functions. The residue frequency-based kernel functions do not need any prior knowledge of selecting the best mutation matrix for use, but the frequency may not be an accurate method for coding biological information in sequences. The homology-based kernel functions are more biological sound, but the determination of the best mutation matrix is not an easy issue. An important research direction is the study of a unified method for designing kernel functions which are both biological sound and less dependent on the prior knowledge of mutation matrix selection.

In both whole sequence and subsequence analysis, positive (functional) class has a much smaller proportion of data. The use of data with unbalanced distribution may result in a model with poor performance. A recent study suggested an alternative way to deal with this issue, where the negative (non-functional) sequences were divided into a couple of subsets, each of which was joined with the same positive sequence set for modelling.[16] Generally, many negative sequences could be redundant. Including the redundant negative sequences for training is wasteful of resources. The other commonly used method in data engineering is random selection that reduces the size of the negative sequences to roughly the same size as the positive sequences.[45] However, this cannot avoid the inclusion of redundant sequences in a data set for modelling. Genetic algorithm and mutual information have therefore been used for selecting the most appropriate training subsequences.[10,39] However, the large diversity in negative sequences prevents the successful use of these methods. The use of the principle of SVMs to learn the generalisation margin

incrementally can be a future research direction for this issue.

CONCLUSION

This review has discussed the applications of support vector machines to the analysis of biological data, mainly focusing on biological sequences, ie protein and DNA sequences. SVMs have also been applied to many other biological data, such as gene expression data.[46,47] There is not enough space here to discuss this important area in detail. However, dealing with sequence data with non-numerical attributes is more of a challenge as most existing coding methods have not been able to code biological information from sequences efficiently.

Three stages of SVM application to DNA and protein sequence analysis

There are in general three stages in using SVMs to analyse protein and DNA sequences. In the first stage, the composition and distributed encoding methods are widely used. In the second stage, HMMs are used to generate profiles for families of sequences. From this, profile features are generated. As the profile method uses only positive data leading to weakened prediction accuracy, homology alignment has been used in the third stage. As the homology alignment method for SVMs still has some difficulties in modelling, more advanced methods, particularly advanced kernel functions, are sought for further improvement of the prediction performance.

Acknowledgments

The author thanks S. Curtis, J. Gan, D. Hoyle, L. Wang, Z. H. Yang and N. Young for their valuable discussions.

References

1. Poorman, R. A., Tomasselli, A. G., Heinrikson, R. L. and Kezdy, F. J. (1991), 'A cumulative specificity model for protease from human immunodeficiency virus types 1 and 2, inferred from statistical analysis of an extended substrate data base', *J. Biol. Chem.*, Vol. 22, pp. 14554–14561.
2. Quinlan, J. R. (1988), 'C4.5; Programs for Machine Learning', Morgan Kaufmann Publishers, San Mateo, CA.
3. Duda, R. O., Hart, P. E. and Stork, D. G. (2002), 'Pattern Classification', 2nd edn, Wiley, Canada.
4. Vapnik, V. (1995), 'The Nature of Statistical Learning Theory', Springer-Verlag, New York.
5. Altschul, S. F., Gish, W., Miller, W. *et al.* (1990), 'Basic Local Alignment Search Tool', *J. Mol. Biol.*, Vol. 215, pp. 403–410.
6. Qian, N. and Sejnowski, T. J. (1988), 'Predicting the secondary structure of globular proteins using neural network models', *J. Mol. Biol.*, Vol. 202, pp. 865–884.
7. Thomson, R., Hodgman, T. C., Yang, Z. R. and Doyle, A. K. (2003), 'Characterising proteolytic cleavage site activity using bio-basis function neural networks', *Bioinformatics*, Vol. 19, pp. 1741–1747.
8. Dayhoff, M. O., Schwartz, R. M. and Orcutt, B. C. (1978), 'A model of evolutionary change in proteins. Matrices for detecting distant relationships', in M. O. Dayhoff, Ed., 'Atlas of Protein Sequence and Structure', Vol. 5, National Biomedical Research Foundation, Washington, DC, pp. 345–358.
9. Johnson, M. S. and Overington, J. P. (1993), 'A structural basis for sequence comparisons – an evaluation of scoring methodologies', *J. Mol. Biol.*, Vol. 233, pp. 716–738.
10. Yang, Z. R. and Thomson, R. (2004), 'Bio-basis function neural network for prediction of protease cleavage sites in proteins', *IEEE Trans Neural Networks* (in press).
11. Cai, Y. D., Ricardo, P. W., Jen, C. H. and Chou, K. C. (2004), 'Application of SVMs to predict membrane protein types', *J. Theoret. Biol.*, Vol. 226, pp. 373–376.
12. Cai, Y. D., Lin, X. J., Xu, X. B. and Chou, K. C. (2002), 'Prediction of protein structural classes by support vector machines', *Computers Chem.*, Vol. 26, pp. 293–296.
13. Hua, S. and Sun, Z. (2001), 'Support vector machine approach for protein subcellular localization prediction', *Bioinformatics*, Vol. 17, pp. 721–728.
14. Chu, F., Jin, G. and Wang, L. (2004), 'Cancer diagnosis and protein secondary structure prediction using support vector machines', in Wang, L., Ed., 'Support Vector Machines, Theory and Applications', Springer-Verlag, Heidelberg.
15. Park, K. and Kanehisa, M. (2003); 'Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs', *Bioinformatics*, Vol. 19, pp. 1656–1663.
16. Carter, R. J., Dubchak, I. and Holbrook, S. R. (2001), 'A computational approach to identify genes for functional RNAs in genomic

sequences', *Nucleic Acids Res.,* Vol. 29, pp. 3928–3938.

17. Ding, C. H. Q. and Dubchak, I. (2001), 'Multi-class protein fold recognition using support vector machines and neural networks', *Bioinformatics*, Vol. 17, pp. 349–358.
18. Cai, C. Z., Wang, W. L., Sun, L. Z. and Chen, Y. Z. (2003), 'Protein function classification via support vector machine approach', *Math. Biosci.*, Vol. 185, pp. 111–122.
19. Cai, Y. D. and Lin, S. L. (2003), 'Support vector machines for predicting rRNA-, RNA-, and DNA-binding proteins from amino acid sequence', *Biochim. Biophys. Acta Proteins Proteomics*, Vol. 1648, pp. 127–133.
20. Lin, K., Kuang, Y., Joseph, J. S. and Kolatkar, P. R. (2002) 'Conserved codon composition of ribosomal protein coding genes in *Escherichia coli*, *Mycobacterium tuberculosis* and *Saccharomyces cerevisiae*: Lessons from supervised machine learning in functional genomics', *Nucleic Acids Res.*, Vol. 30, pp. 2599–2607.
21. Jaakkola, T., Diekhans, M. and Haussler, D. (1999), 'Using the Fisher kernel method to detect remote protein homologies', *Proc. Int. Conf. Intelligent Systems Mol. Biol.*, Vol. 7, pp. 149–158.
22. Jaakkola, T., Diekhans, M. and Haussler, D. (2000), 'A discriminative framework for detecting remote protein homologies', *J. Comput. Biol.*, Vol. 7, pp. 95–114.
23. Karchin, R., Karplus, K. and Haussler, D. (2002), 'Classifying G-protein coupled receptors with support vector machines', *Bioinformatics*, Vol. 18, pp. 147–159.
24. Guermeur, Y., Pollastri, G., Elisseeff, A. *et al.* (2004), 'Combining protein secondary structure prediction models with ensemble methods of optimal complexity', *Neurocomputing,* Vol. 56, pp. 305–327.
25. Liao, L. and Noble, W.S. (2002), 'Combining pairwise sequence homology and support vector machines for remote protein homology detection', *Proc. Int. Conf. Comput. Mol. Biol.*, Vol. 6, pp. 225–232.
26. Liao, L. and Noble, W. S. (2003), 'Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships', *J. Comput. Biol.*, Vol. 10, pp. 857–868.
27. Leslie, C. S., Eskin, E., Cohen, A. *et al.* (2004), 'Mismatch string kernels for discriminative protein classification', *Bioinformatics*, Vol. 20, 467–476.
28. Ward, J. J., Sodhi, J. S., McGuffin, L. J. *et al.* (2004), 'Prediction and functional analysis of native disorder in proteins from the three kingdoms of life', *J. Mol. Biol.*, Vol. 337, pp. 635–645.
29. Saigo, H., Vert, J. P., Ueda, N. and Akutsu, T. (2004) Protein homology detection using string alignment kernels', *Bioinformatics* (in press).
30. Zien, A., Ratsch, G., Mika, S. *et al.* (2000), 'Engineering support vector machine kernels that recognize translation initiation sites', *Bioinformatics*, Vol. 16, pp. 799–807.
31. Zavaljevski, N., Stevens, F. J. and Reifman, J. (2002), 'Support vector machines with selective kernel scaling for protein classification and identification of key amino acid positions', *Bioinformatics*, Vol. 18, pp. 689–696.
32. Cai, Y. D., Feng, K. Y., Li, Y. X. and Chou, K. C. (2003), 'Support vector machine for predicting alpha-turn types', *Peptides*, Vol. 24, pp. 629–630.
33. Kim, J. H., Lee, J., Oh, B. *et al.* (2004), 'Prediction of phosphorylation sites using SVMs', *Bioinformatics* (in press).
34. Zhao, Y., Pinilla, C., Valmori, D. *et al.* (2003), 'Application of support vector machines for T-cell epitopes prediction', *Bioinformatics*, Vol. 19, pp. 1978–1984.
35. Koike, A. and Takagi, T. (2004), 'Prediction of protein–protein interaction sites using support vector machines', *Protein Eng. Des. Sel.*, Vol. 17, pp. 165–173.
36. Yang, Z. R. and Berry, E. (2004) 'Reduced bio-basis function neural networks for protease cleavage site prediction', *J. Comput. Biol. Bioinformatics* (in press).
37. Sidhu, A. and Yang, Z. R. (2004), 'Prediction of signal peptides using bio-basis function neural networks and decision tree method', *Appl. Bioinformatics* (in press).
38. Thomson, R. and Esnouf, R. (2004), 'Prediction of natively disordered regions in proteins using a bio-basis function neural network', *Lecture Notes Comp. Sci.*, Vol. 3177, pp. 109–117.
39. Berry, E., Dalby, A. and Yang, Z. R. (2004), 'Reduced bio-basis function neural networks in prediction of phosphorylation sites, a comparative study', *Comput. Biol. Chem.*, Vol. 28, pp. 75–85.
40. Yang, Z. R. and Chou, K. C. (2004), 'Bio-basis function neural networks for the prediction of the O-linkage sites in glyco-proteins', *Bioinformatics*, Vol. 20, pp. 903–908.
41. Yang, Z. R. and Chou, K. C. (2003), 'Bio-support vector machines for computational proteomics', *Bioinformatics*, Vol. 19, pp. 1–7.
42. Cai, Y. D., Ricardo, P. W., Jen, C. H. and Chou, K. C. (2004), 'Application of SVM to predict membrane protein types', *J. Theoret. Biol.*, Vol. 226, pp. 373–376.
43. Dobson, P. D. and Doig, A. J. (2003), 'Distinguishing enzyme structures from non-

enzymes without alignments', *J. Mol. Biol.*, Vol. 330, pp. 771–783.

44. Metz, C. E. (1978), 'Basic principles of ROC analysis', *Seminars in Nuclear Medicine*, Vol. 8, pp. 283–298.
45. Wilson, R. L. and Sharda, R. (1994), 'Bankruptcy prediction using neural networks', *Decision Support Systems*, Vol. 11, pp. 545–557.
46. Brown, M. P. S., Grundy, W. N., Lin, D. *et al.* (2000), 'Knowledge-based analysis of microarray gene expression data by using support vector machines', *Proc. Natl Acad. Sci. USA*, Vol. 97, pp. 262–267.
47. Furey, T. S., Cristianini, N., Duffy, N. *et al.* (2000), 'Support vector machine classification and validation of cancer tissue samples using microarray expression data', *Bioinformatics*, Vol. 16, pp. 906–914.