Support Vector Machines: Theory and Applications

Lipo Wang

(ed.)

Springer, Berlin

2005

Preface

The support vector machine (SVM) is a supervised learning method that generates input-output mapping functions from a set of labeled training data. The mapping function can be either a classification function, i.e., the category of the input data, or a regression function. For classification, nonlinear kernel functions are often used to transform input data to a high-dimensional feature space in which the input data become more separable compared to the original input space. Maximum-margin hyperplanes are then created. The model thus produced depends on only a subset of the training data near the class boundaries. Similarly, the model produced by Support Vector Regression ignores any training data that is sufficiently close to the model prediction. SVMs are also said to belong to "kernel methods".

In addition to its solid mathematical foundation in statistical learning theory, SVMs have demonstrated highly competitive performance in numerous real-world applications, such as bioinformatics, text mining, face recognition, and image processing, which has established SVMs as one of the state-of-the-art tools for machine learning and data mining, along with other soft computing techniques, e.g., neural networks and fuzzy systems.

This volume is composed of 20 chapters selected from the recent myriad of novel SVM applications, powerful SVM algorithms, as well as enlightening theoretical analysis. Written by experts in their respective fields, the first 12 chapters concentrate on SVM theory, whereas the subsequent 8 chapters emphasize practical applications, although the "decision boundary" separating these two categories is rather "fuzzy".

Kecman first presents an introduction on the SVM, explaining the basic theory and implementation aspects. In the chapter contributed by Ma and Cherkassky, a novel approach to nonlinear classification using a collection of several simple (linear) classifiers is proposed based on a new formulation of the learning problem called multiple model estimation. Pelckmans, Goethals, De Brabanter, Suykens, and De Moor describe componentwise Least Squares Support Vector Machines (LS-SVMs) for the estimation of additive models consisting of a sum of nonlinear components.

Motivated by the statistical query model, Mitra, Murthy and Pal study an active learning strategy to solve the large quadratic programming problem of SVM design in data mining applications. Kaizhu Huang, Haiqin Yang, King, and Lyu propose a unifying theory of the Maxi-Min Margin Machine (M4) that subsumes the SVM, the minimax probability machine, and the linear discriminant analysis. Vogt and Kecman present an active-set algorithm for quadratic programming problems in SVMs, as an alternative to working-set (decomposition) techniques, especially when the data set is not too large, the problem is ill-conditioned, or when high precision is needed.

Being aware of the abundance of methods for SVM model selection, Anguita, Boni, Ridella, Rivieccio, and Sterpi carefully analyze the most well-known methods and test some of them on standard benchmarks to evaluate their effectiveness. In an attempt to minimize bias, Peng, Heisterkamp, and Dai propose locally adaptive nearest neighbor classification methods by using locally linear SVMs and quasiconformal transformed kernels. Williams, Wu, and Feng discuss two geometric methods to improve SVM performance, i.e., (1) adapting kernels by magnifying the Riemannian metric in the neighborhood of the boundary, thereby increasing class separation, and (2) optimally locating the separating boundary, given that the distributions of data on either side may have different scales.

Song, Hu, and Xulei Yang derive a Kuhn-Tucker condition and a decomposition algorithm for robust SVMs to deal with overfitting in the presence of outliers. Lin and Sheng-de Wang design a fuzzy SVM with automatic determination of the membership functions. Kecman, Te-Ming Huang, and Vogt present the latest developments and results of the Iterative Single Data Algorithm for solving large-scale problems.

Exploiting regularization and subspace decomposition techniques, Lu, Plataniotis, and Venetsanopoulos introduce a new kernel discriminant learning method and apply the method to face recognition. Kwang In Kim, Jung, and Hang Joon Kim employ SVMs and neural networks for automobile license plate localization, by classifying each pixel in the image into the object of interest or the background based on localized color texture patterns. Mattera discusses SVM applications in signal processing, especially the problem of digital channel equalization. Chu, Jin, and Lipo Wang use SVMs to solve two important problems in bioinformatics, i.e., cancer diagnosis based on microarray gene expression data and protein secondary structure prediction.

Emulating the natural nose, Brezmes, Llobet, Al-Khalifa, Maldonado, and Gardner describe how SVMs are being evaluated in the gas sensor community to discriminate different blends of coffee, different types of vapors and nerve agents. Zhan presents an application of the SVM in inverse problems in ocean color remote sensing. Liang uses SVMs for non-invasive diagnosis of delayed gastric emptying from the cutaneous electrogastrograms (EGGs). Rojo-Álvarez, García-Alberola, Artés-Rodríguez, and Arenal-Maíz apply SVMs, together with bootstrap resampling and principal component analysis, to tachycardia discrimination in implantable cardioverter defibrillators.

I would like to express my sincere appreciation to all authors and reviewers who have spent their precious time and efforts in making this book a reality. I wish to especially thank Professor Vojislav Kecman, who graciously took on the enormous task of writing a comprehensive introductory chapter, in addition to his other great contributions to this book. My gratitude also goes to Professor Janusz Kacprzyk and Dr. Thomas Ditzinger for their kindest support and help with this book.

Singapore
January 2005

Lipo Wang

Contents

Cancer Diagnosis and Protein Secondary Structure Prediction Using Support Vector Machines

F. Chu, G. Jin, and L. Wang

School of Electrical and Electronic Engineering,
Nanyang Technological University,
Block S1, Nanyang Avenue, Singapore, 639798
elpwang@ntu.edu.sg

Abstract. In this chapter, we use support vector machines (SVMs) to deal with two bioinformatics problems, i.e., cancer diagnosis based on gene expression data and protein secondary structure prediction (PSSP). For the problem of cancer diagnosis, the SVMs that we used achieved highly accurate results with fewer genes compared to previously proposed approaches. For the problem of PSSP, the SVMs achieved results comparable to those obtained by other methods.

Key words: support vector machine, cancer diagnosis, gene expression, protein secondary structure prediction

1 Introduction

Support Vector Machines (SVMs) [1, 2, 3] have been widely applied to pattern classification problems [4, 5, 6, 7, 8] and nonlinear regressions [9, 10, 11]. In this chapter, we apply SVMs to two pattern classification problems in bioinformatics. One is cancer diagnosis based on microarray gene expression data; the other is protein secondary structure prediction (PSSP). We note that the meaning of the term *prediction* is different from that in some other disciplines, e.g., in time series prediction where prediction means guessing future trends from past information. In PSSP, "prediction" means supervised classification that involves two steps. In the first step, an SVM is trained as a classifier with a part of the data in a specific protein sequence data set. In the second step (i.e., prediction), we use the classifier trained in the first step to classify the rest of the data in the data set.

In this work, we use the C-Support Vector Classifier (C-SVC) proposed by Cortes and Vapnik [1] available in the LIBSVM library [12]. The C-SVC has radial basis function (RBF) kernels. Much of the computation is spent on

F. Chu, G. Jin, and L. Wang: *Cancer Diagnosis and Protein Secondary Structure Prediction Using Support Vector Machines*, StudFuzz **177**, 343–363 (2005)
www.springerlink.com

tuning two important parameters, i.e., γ and C. γ is the parameter related to the span of an RBF kernel: the smaller the value is, the wider the kernel spans. C controls the tradeoff between the complexity of the SVM and the number of nonseparable samples. A larger C usually leads to higher training accuracy. To achieve a good performance, various combinations of the pair (C, γ) have to be tested, ideally, to find the optimal combination.

This chapter is organized as follows. In Sect. 2, we apply SVMs to cancer diagnosis with microarray data. In Sect. 3, we review the PSSP problem and its biological background. In Sect. 4, we apply SVMs to the PSSP problem. In the last section, we draw our conclusions.

2 SVMs for Cancer Type Prediction

Microarrays [15, 16] are also called gene chips or DNA chips. On a microarray chip, there are thousands of spots. Each spot contains the clone of a gene from one specific tissue. At the same time, some mRNA samples are labelled with two different kinds of dyes, for example, Cy5 (red) and Cy3 (blue). After that, the mRNA samples are put on the chip and interact with the genes on the chip. This process is called *hybridization*. The color of each spot on the chip changes after hybridization. The image of the chip is then scanned out and reflects the characteristics of the tissue at the molecular level. Using microarrays for different tissues, biological and biomedical researchers are able to compare the difference of those tissues at the molecular level. Figure 1 summarizes the process of making microarrays.

In recent years, cancer type/subtype prediction has drawn a lot of attention in the context of the microarray technology that is able to overcome some limitations of traditional methods. Traditional methods for diagnosis of different types of cancers are mainly based on morphological appearances

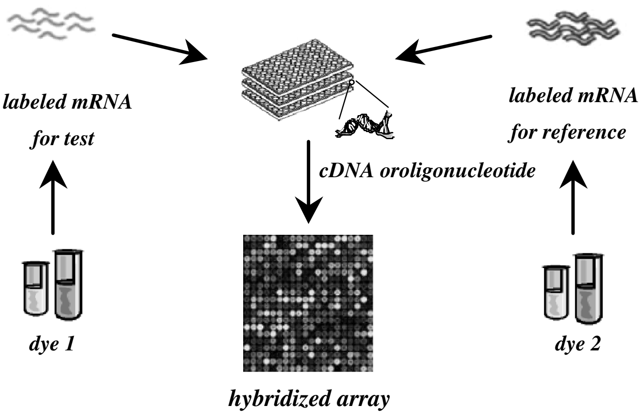


Fig. 1. The process of making microarrays

of cancers. However, sometimes it is extremely difficult to find clear distinctions between some types of cancers according to their appearances. Thus, the newly appeared microarray technology is naturally applied to this muddy problem. In fact, gene-expression-based cancer classifiers have achieved good results in classifying lymphoma [17], leukemia [18], breast cancer [19], liver cancer [20], and so on.

Gene-expression-based cancer classification is challenging due to the following two properties of gene expression data. Firstly, gene expression data are usually very high dimensional. The dimensionality usually ranges from several thousands to over ten thousands. Secondly, gene expression data sets usually contain relatively small numbers of samples, e.g., a few tens. If we treat this pattern recognition problem with supervised machine learning approaches, we need to deal with the shortage of training samples and high dimensional input features.

Recent approaches to this problem include artificial neural networks [21], an evolutionary algorithm [22], nearest shrunken centroids [23], and a graphical method [24]. Here, we use SVMs to solve this problem.

2.1 Gene Expression Data Sets

In the following parts of this section, we describe three data sets to be used in this chapter. One is the small round blue cell tumors (SRBCTs) data set [21]. Another is the lymphoma data set [17]. The last one is the leukemia data set [18].

The SRBCT Data Set

The SRBCT data set (http://research.nhgri.nih.gov/microarray/Supplement/) [21] includes the expression data of 2308 genes. Khan et al. provided totally 63 training samples and 25 testing samples, five of the testing samples being not SRBCTs. The 63 training samples contain 23 Ewing family of tumors (EWS), 20 rhabdomyosarcoma (RMS), 12 neuroblastoma (NB), and 8 Burkitt lymphomas (BL). And the 20 SRBCTs testing samples contain 6 EWS, 5 RMS, 6 NB, and 3 BL.

The Lymphoma Data Set

The lymphoma data set (http://llmpp.nih.gov/lymphoma) [17] has 62 samples in total. Among them, 42 samples are derived from diffuse large B-cell lymphoma (DLBCL), 9 samples from follicular lymphoma (FL), and 11 samples from chronic lymphocytic lymphoma (CLL). The entire data set includes the expression data of 4026 genes. We randomly divided the 62 samples into two parts, 31 for training and the other 31 for testing. In this data set, a small part of data is missing. We applied a k-nearest neighbor algorithm [25] to fill those missing values.

The Leukemia Data Set

The leukemia data set (www-genome.wi.mit.edu/MPR/data_set_ALL_AML.html) [18] contains two types of samples, i.e. the acute myeloid leukemia (AML) and the acute lymphoblastic leukemia (ALL). Golub et al. provided 38 training samples and 34 testing samples. The entire leukemia data set contains the expression data of 7129 genes.

Ordinarily, raw gene expression data should be normalized to reduce the systemic bias introduced during experiments. For the SRBCT and the lymphoma data sets, normalized data can be found on the web. However, for the leukemia data set, such normalized data are not available. Thereafter, we need to do normalization ourselves.

We followed the normalization procedure used in [26]. Three steps were taken, i.e., (a) setting threshold with a floor of 100 and a ceiling of 16000, that is, if a value is greater (smaller) than the ceiling (floor), this value is replaced by the ceiling (floor); (b) filtering, leaving out the genes with $\max / \min \leq 5$ or $(\max - \min) \leq 500$ (max and min refer to the maximum and minimum of the expression values of a gene, respectively); (c) carrying out logarithmic transformation with 10 as the base to all the expression values. 3571 genes survived after these three steps. Furthermore, the data were standardized across experiments, i.e., subtracted by the mean and divided by the standard deviation of each experiment.

2.2 A T-Test-Based Gene Selection Approach

The t-test is a statistical method proposed by Welch [27] to measure how large the difference is between the distributions of two groups of samples. If a gene shows large distinctions between 2 groups, the gene is important for classification of the two groups. To find the genes that contribute most to classification, t-test has been used in gene selection [28] in recent years.

Selecting important genes using t-test involves several steps. In the first step, a score based on the t-test (named t-score or TS) is calculated for each gene. In the second step, all the genes are rearranged according to their TSs. The gene with the largest TS is put in the first place of the ranking list, followed by the gene with the second largest TS, and so on.

Finally, only some top genes in the list are used for classification. The standard t-test is applicable to measure the difference between only two groups. Therefore, when the number of classes is more than two, we need to modify the standard t-test. In this case, we use the t-test to measure the difference between one specific class and the centroid of all the classes. Hence, the definition of the TS for gene i can be described as follows:

$$TS_i = \max\left\{\left|\frac{\overline{x}_{ik} - \overline{x}_i}{m_k s_i}\right| \ , \quad k = 1, 2, \ldots K\right\} \tag{1}$$

$$\overline{x}_{ik} = \sum_{j \in C_k} \overline{x}_{ij} / n_k \tag{2}$$

$$\overline{x}_i = \sum_{j=1}^{n} x_{ij} / n \tag{3}$$

$$s_i^2 = \frac{1}{n - K} \sum_k \sum_{j \in C_k} (x_{ij} - \overline{x}_{ik})^2 \tag{4}$$

$$m_k = \sqrt{1/n_k + 1/n} \tag{5}$$

There are K classes. $\max\{y_k, k = 1, 2, \ldots K\}$ is the maximum of all y_k. C_k refers to class k that includes n_k samples. x_{ij} is the expression value of gene i in sample j. $\overline{x}_{ik}$ is the mean expression value in class k for gene i. n is the total number of samples. $\overline{x}_i$ is the general mean expression value for gene i. s_i is the pooled within-class standard deviation for gene i.

2.3 Experimental Results

We applied the above gene selection approach and the C-SVC to process the SRBCT, the lymphoma, and the leukemia data sets.

Results for the SRBCT Data Set

In the SRBCT data set, we firstly ranked the importance of all the genes with TSs. We picked out 60 of the genes with the largest TSs to do classification. The top 30 genes are listed in Table 1. We input these genes one by one to the SVM classifier according to their ranks. That is, we first input the gene ranked No.1 in Table 1. Then, we trained the SVM classifier with the training data and tested the SVM classifier with the testing data. After that, we repeated the whole process with the top 2 genes in Table 1, and then the top 3 genes, and so on. Figure 2 shows the training and the testing accuracies with respect to the number of genes used.

In this data set, we used SVMs with RBF kernels. C and γ were set as 80 and 0.005, respectively. This classifier obtained 100% training accuracy and 100% testing accuracy using the top 7 genes. In fact, the values of C and γ have great impact on the classification accuracy. Figure 3 shows the classification results with different values of γ. We also applied SVMs with linear kernels (with kernel function $K(\mathbf{X}, \mathbf{X}_i) = \mathbf{X}^T\mathbf{X}_i$) and SVMs with polynomial kernels (with kernel function $K(\mathbf{X}, \mathbf{X}_i) = (\mathbf{X}^T\mathbf{X}_i + 1)^p$ and order $p = 2$) to the SRBCT data set. The results are shown in Fig. 4 and Fig. 5. The SVMs with linear kernels and the SVMs with polynomial kernels obtained 100% accuracy with 7 and 6 genes, respectively. The similarity of these results indicates that the SRBCT data set is separable for all the three kinds of SVMs.

Table 1. The 30 top genes selected by the t-test in the SRBCT data set

Rank	Gene ID	Gene Description
1	810057	cold shock domain protein A
2	784224	fibroblast growth factor receptor 4
3	296448	insulin-like growth factor 2 (somatomedin A)
4	770394	Fc fragment of IgG, receptor, transporter, alpha
5	207274	Human DNA for insulin-like growth factor II (IGF-2); exon 7 and additional ORF
6	244618	ESTs
7	234468	ESTs
8	325182	cadherin 2, N-cadherin (neuronal)
9	212542	Homo sapiens mRNA; cDNA DKFZp586J2118 (from clone DKFZp586J2118)
10	377461	caveolin 1, caveolae protein, 22 kD
11	41591	meningioma (disrupted in balanced translocation) 1
12	898073	transmembrane protein
13	796258	sarcoglycan, alpha (50kD dystrophin-associated glycoprotein)
14	204545	ESTs
15	563673	antiquitin 1
16	44563	growth associated protein 43
17	866702	protein tyrosine phosphatase, non-receptor type 13 (APO-1/CD95 (Fas)-associated phosphatase)
18	21652	catenin (cadherin-associated protein), alpha 1 (102 kD)
19	814260	follicular lymphoma variant translocation 1
20	298062	troponin T2, cardiac
21	629896	microtubule-associated protein 1 B
22	43733	glycogenin 2
23	504791	glutathione S-transferase A4
24	365826	growth arrest-specific 1
25	1409509	troponin T1, skeletal, slow
26	1456900	Nil
27	1435003	tumor necrosis factor, alpha-induced protein 6
28	308231	Homo sapiens incomplete cDNA for a mutated allele of a myosin class I, myh-1c
29	241412	E74-like factor 1 (ets domain transcription factor)
30	1435862	antigen identified by monoclonal antibodies 12E7, F21 and O13

For the SRBCT data set, Khan et al. [21] 100% accurately classified the 4 types of cancers with a linear artificial neural network by using 96 genes. Their results and our results of the linear SVMs both proved that the classes in the SRBCT data set are linearly separable. In 2002, Tibshirani et al. [23] also correctly classified the SRBCT data set with 43 genes by using a method named nearest shrunken centroids. Deutsch [22] further reduced the number of genes required for reliable classification to 12 with an evolutionary algorithm. Compared with these previous results, the SVMs that we used can achieve

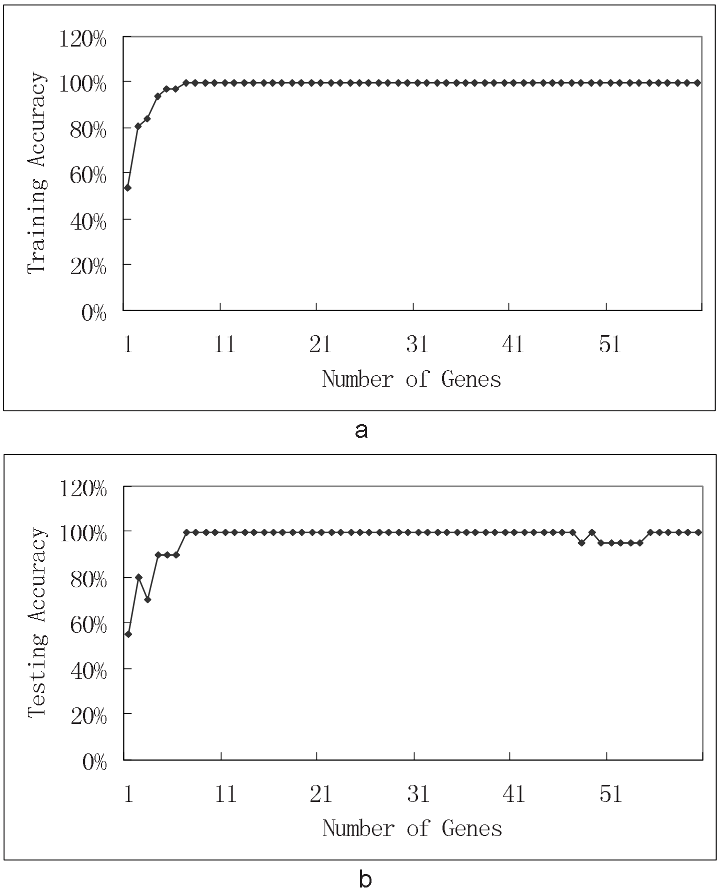


Fig. 2. The classification results vs. the number of genes used for the SRBCT data set: (**a**) the training accuracy; (**b**) the testing accuracy

100% accuracy with only 6 genes (for the polynomial kernel function version, $p = 2$) or 7 genes (for the linear and the RBF kernel function versions). Table 2 summarizes this comparison.

Results for the Lymphoma Data Set

In the lymphoma data set, we selected the top 70 genes. The training and testing accuracies with the 70 top genes are shown in Fig. 6. The classifiers used here are also SVMs with RBF kernels. The best C and γ obtained are equal to 20 and 0.1, respectively. The SVMs obtained 100% accuracy for both the training and testing data with only 5 genes.

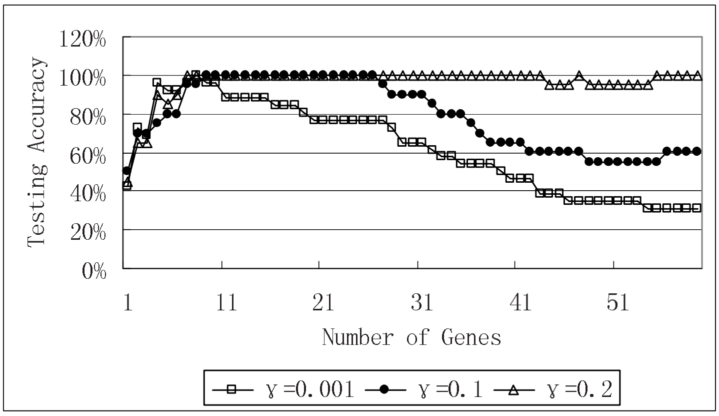


Fig. 3. The testing results of SVMs with RBF kernels and different values of γ for the SRBCT data

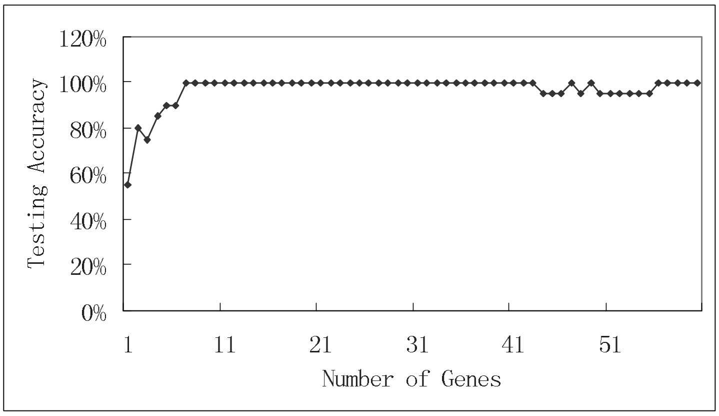


Fig. 4. The testing results of the SVMs with linear kernels for the SRBCT data

For the lymphoma data set, nearest shrunken centroids [29] used 48 genes to give a 100% accurate classification. In comparison with this, the SVMs that we used greatly reduced the number of genes required.

Table 2. Comparison of the numbers of genes required by different methods to achieve 100% classification accuracy

Method	Number of Genes Required
Linear MLP neural network [21]	96
Nearest shrunken centroids [23]	43
Evolutionary algorithm [22]	12
SVM (linear or RBF kernel function)	7
SVM (polynomial kernel function, $p = 2$)	6

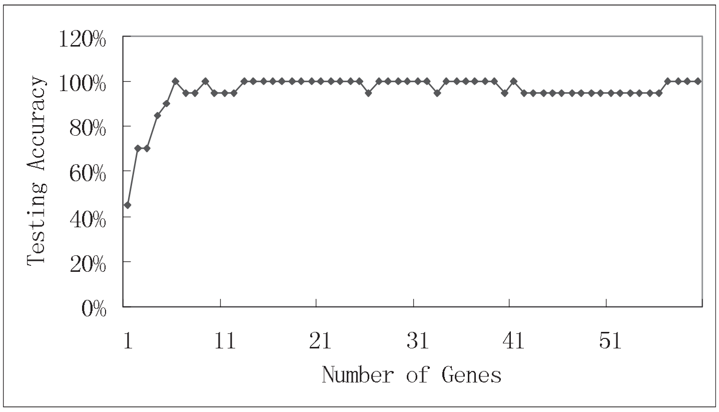


Fig. 5. The testing result of the SVMs with polynomial kernels ($p = 2$) for the SRBCT data

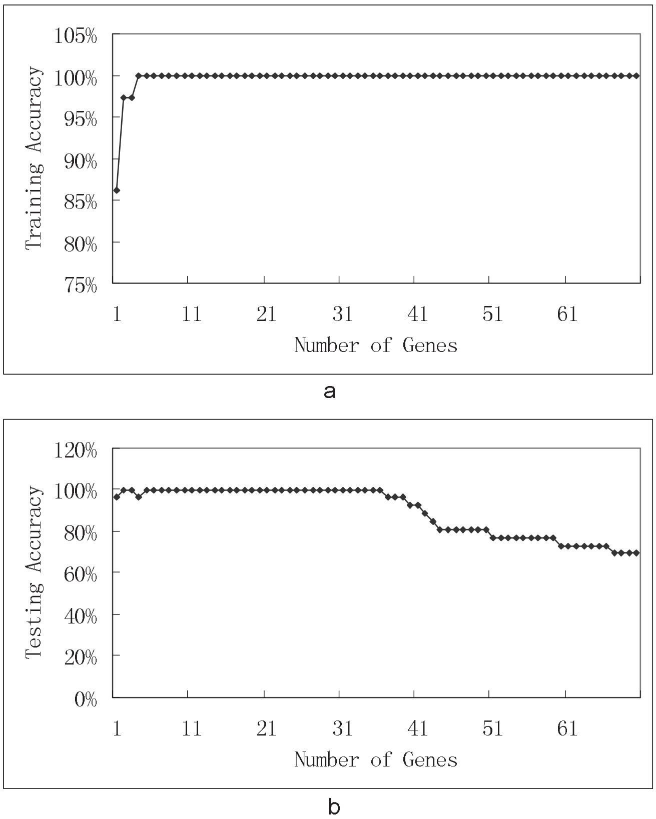


Fig. 6. The classification results vs. the number of genes used for the lymphoma data set: (**a**) the training accuracy; (**b**) the testing accuracy

Results for the Leukemia Data Set

Alizadeh et al. [17] built a 50-gene classifier that made 1 error in the 34 testing samples; and in addition, it cannot give strong prediction to another 3 samples. Nearest shrunken centroids made 2 errors among the 34 testing samples with 21 genes [23]. As shown in Fig. 7, we used the SVMs with RBF kernels with 2 errors for the testing data but with only 20 genes.

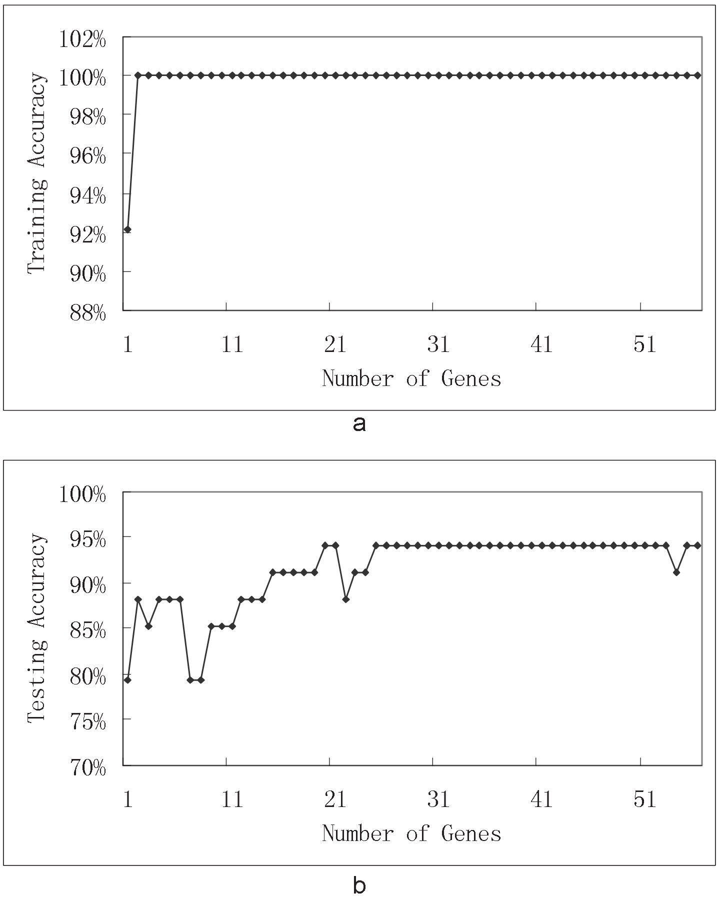
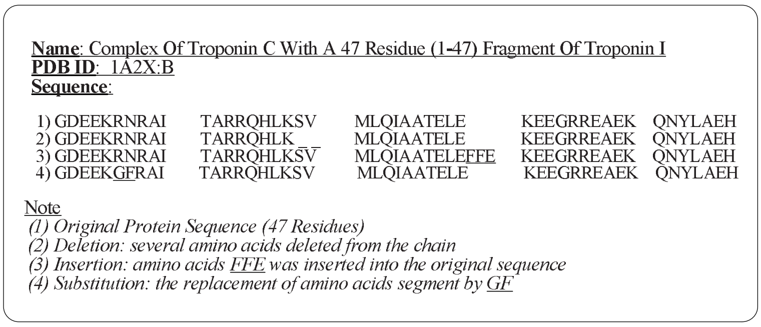


Fig. 7. The classification results vs. the number of genes used for the leukemia data set: (**a**) the training accuracy; (**b**) the testing accuracy



Name: Complex Of Troponin C With A 47 Residue (1-47) Fragment Of Troponin I
PDB ID: 1A2X:B
Sequence:

```
 1) GDEEKRNRAI    TARRQHLKSV    MLQIAATELE      KEEGRREAEK   QNYLAEH
 2) GDEEKRNRAI    TARRQHLK _ _  MLQIAATELE      KEEGRREAEK   QNYLAEH
 3) GDEEKRNRAI    TARRQHLKSV    MLQIAATELEFFE   KEEGRREAEK   QNYLAEH
 4) GDEEKGFRAI    TARRQHLKSV    MLQIAATELE      KEEGRREAEK   QNYLAEH
```

Note
(1) Original Protein Sequence (47 Residues)
(2) Deletion: several amino acids deleted from the chain
(3) Insertion: amino acids FFE was inserted into the original sequence
(4) Substitution: the replacement of amino acids segment by GF

Fig. 8. An example of alphabetical representations of protein sequences and protein mutations. PDB stands for protein data bank [30]

3 Protein Secondary Structure Prediction

3.1 The Biological Background of the PSSP

A protein sequence is a linear array of amino acids. Each amino acid consists of 3 consecutively ordered DNA bases (A, T, C, or G). An amino acid carries various kinds of information determined by its DNA combination. An amino acid is a basic unit of a protein sequence and is called a *residue*. There are altogether 20 types of amino acids and each type of amino acids is denoted by an English character. For example, the character “A” is used to represent the type of amino acid named Alanine. Thus, a protein sequence in the alphabetical representation is a long sequence of characters, as shown in Fig. 8. Given a protein sequence, various evolutionary environments may induce mutations, including insertions, deletions, or substitutions, to the original protein, thereby producing diversified yet biologically similar organisms.

3.2 Types of Protein Secondary Structures

Secondary structures are formed by hydrogen bonds between relatively small segments of protein sequences. There are three common secondary structures in proteins, namely *α-helix, β-sheet (strand)* and *coil.*

Figure 9 visualizes protein secondary structures. In Fig. 9, the dark ribbons represent helices and the gray ribbons are sheets. And the strings in between are coils that bind helices and sheets.

3.3 The Task of PSSP

In the context of PSSP, “prediction” carries similar meaning as that of classification: given a residue of a protein sequence, the predictor should classify the residue into one of the three secondary structure states according to the residue’s characteristics. PSSP is usually conducted in two stages: sequence-structure (Q2T) prediction and structure-structure (T2T) prediction.

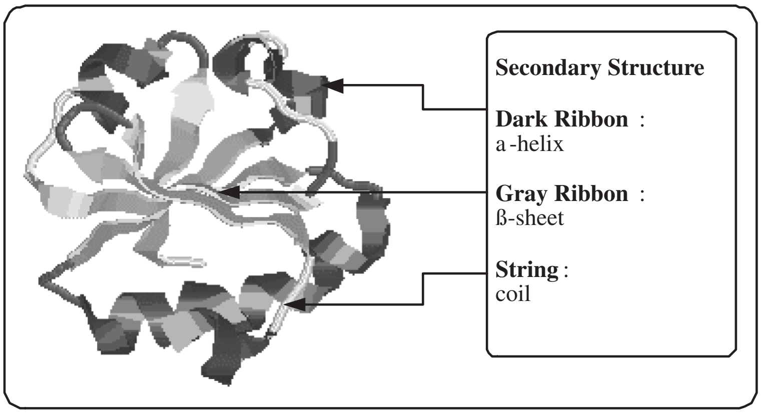


Fig. 9. Three types of protein secondary structures: α-helix, β-strand, and coil

Sequence-Structure (Q2T) Prediction

Q2T prediction predicts the protein secondary structure from protein sequences. Given a protein sequence, a Q2T predictor maps each residue of the sequence to a relevant secondary structure state by inspecting the distinct characteristics of the residue, e.g., the type of the amino acid, the sequence context (that is, what are the neighboring residues), and evolutionary information. The sequence-structure prediction plays the most important role in PSSP.

Structure-Structure (T2T) Prediction

For common pattern classification problems, it would be the end of the task once each data point (each residue in our case) has been assigned a class label. Classification usually do not continue to a second phase. However, the problem we are dealing with is different from most pattern recognition problems. In a typical pattern classification problem, the data points are assumed to be independent. But this is not true for the PSSP problem because the neighboring sequence positions usually provide some meaningful information. For example, an α-helix usually consists of at least 3 consecutive residues of the same secondary structure state (e.g., $\ldots\alpha\alpha\alpha\ldots$). Therefore, if an alternative occurrence of the α-helix and the β-strand (e.g., $\ldots\alpha\beta\alpha\beta\ldots$) is predicted, it would be incorrect. Thus, T2T prediction based on the Q2T results is usually carried out. This step helps to correct errors incurred in Q2T prediction and hence enhances the overall prediction accuracy. Figure 10 illustrates PSSP with the two stages.

Note that amino acids of the same type do not always have the same secondary structure state. For instance, in Fig. 10, the *12*-th and the *20*-th amino residues counted from the left side are both F. However, they are assigned to two different secondary structure states, i.e., α and β.

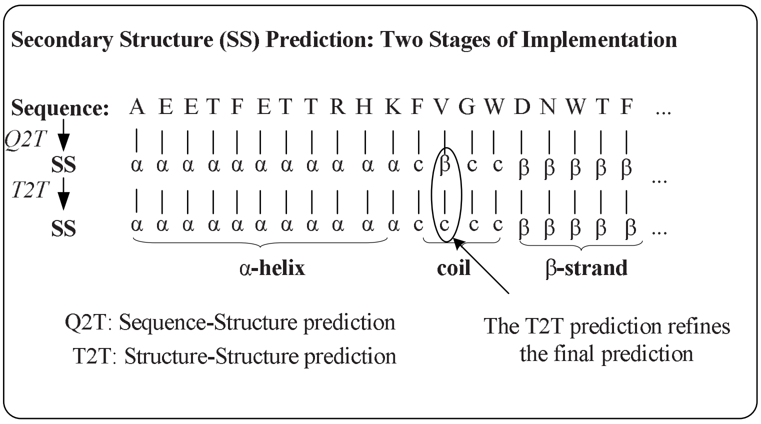

Secondary Structure (SS) Prediction: Two Stages of Implementation
Sequence: A E E T F E T T R H K F V G W D N W T F ...
Q2T
SS α α α α α α α α α α α c β c c β β β β β ...
T2T
SS α α α α α α α α α α α c c c c β β β β β ...
α-helix
coil
β-strand
Q2T: Sequence-Structure prediction
T2T: Structure-Structure prediction
The T2T prediction refines the final prediction

Fig. 10. Protein secondary structure prediction: the two-stage approach

Prediction of the secondary structure state at each sequence position should not solely rely on the residue at that position. A window expanding towards both directions of the residue should be used to include the sequence context.

3.4 Methods for PSSP

PSSP was stimulated by research on protein 3D structures in the 1960s [31, 32], which attempted to find the correlations between protein sequences and secondary structures. This was the first generation of PSSP, where most methods carried out prediction based on single residue statistics [33, 34, 35, 36, 37]. Since only particular types of amino acids from protein sequences were extracted and used in experiments, the accuracies of these methods were more or less over-estimated [38].

With growth of knowledge on protein structures, the second generation PSSP made use of segment statistics. A segment of residues was studied to find out how likely the central residue of the segment belonged to a secondary structure state. Algorithms of this generation include statistical information [36, 40], sequence patterns [41, 42], multi-layer networks [43, 44, 45, 49], multivariate statistics [46], nearest-neighbor algorithms [47], etc.

Unfortunately, the methods in both the first and the second generations could not reach an accuracy higher than 70%.

The earliest application of artificial neural networks to PSSP was carried out by Qian and Sejnowski in 1988 [48]. They used a three-layered back-propagation network whose input data was encoded with a scheme called BIN21. Under BIN21, each input data was a sliding window of 13 residues obtained by extending 6 sequence positions from the central residue. The focus of each observation was only on the central residue, i.e., only the central residue was assigned to one of the three possible secondary structure states

(α-helix, β-strand, and coil). Modifications to the BIN21 scheme were introduced in two later studies. Kneller et al. [49] added one additional input unit to present the hydrophobicity scale of each amino acid residue and showed a slightly higher accuracy. Sasagawa and Tajima [51] used the BIN24 scheme to encode three additional amino acid alphabets, B, X, and Z. The above early work had an accuracy ceiling of 65%. In 1995, Vivarelli et al. [52] used a hybrid system that combined a Local Genetic Algorithm (LGA) and neural networks for PSSP. Although LGA was able to select network topologies efficiently, it still could not break through the accuracy ceiling, regardless of the network architectures applied.

A significant improvement of the 3-state secondary structure prediction came from Rost and Sander's method (PHD) [53, 54], which was based on a multi-layer back-propagation network. Different from the BIN21 coding scheme, PHD took into account evolutionary information in the form of *multiple sequence alignments* to represent the input data. This inclusion of the protein family information improved the prediction accuracy by around six percentages. Moreover, another cascaded neural network conducted structure-structure prediction. Using the 126 protein sequences (RS126) developed by themselves, Rost and Sander achieved the overall accuracy as high as 72%.

In 1999, Jones [56] used a Position-Specific Scoring Matrix (PSSM) [57, 58] obtained from the online alignment searching tool PSI-Blast (http://www.ncbi.nlm.nih.gov/BLAST/) to numerically represent the protein sequence. A PSSM was constructed automatically from a multiple alignment of the highest scoring hits in an initial BLAST search. The PSSM was generated by calculating position-specific scores for each position in the alignment. Highly conserved positions of protein sequence received high scores and weakly conserved positions received scores near zero. Due to its high accuracy in finding the biologically similar protein sequences, the evolutionary information carried by the PSSM is more sensitive than the profiles obtained by other multiple sequence alignment approaches. With a neural network similar to that of Rost and Sander's, Jones' PSIPRED method achieved an accuracy as high as 76.5% using a much larger data set than RS126.

In 2001, Hua and Sun [6] proposed an SVM approach. This was an early application of the SVM to the PSSP problem. In their work, they first constructed 3 one-versus-one and 3 one-versus-all binary classifiers. Three tertiary classifiers were designed based on these binary classifiers through the use of the largest response, the decision tree and votes for the final decision. By making use of the Rost's data encoding scheme, they achieved the accuracy of 71.6% and the segment overlap accuracy of 74.6% for the RS126 data set.

4 SVMs for the PSSP Problem

In this section, we use the LIBSVM, or more specially, the C-SVC, to solve the PSSP problem.

The data set used here was originally developed and used by Jones [56]. This data set can be obtained from the website (http://bioinf.cs.ucl.ac.uk/psipred/). The data set contains a total of 2235 protein sequences for training and 187 sequences for testing. All the sequences in this data set have been processed by the online alignment searching tool PSI-Blast (http://www.ncbi.nlm.nih.gov/BLAST/).

As mentioned above, we will conduct PSSP in two stages, i.e., Q2T prediction and T2T prediction.

4.1 Q2T Prediction

Parameter Tuning Strategy

For PSSP, there are three parameters, i.e., the window size N, and SVM parameters (C,γ), to be tuned. N determines the span of the sliding window, i.e., how many neighbors to be included in the window. Here, we test four different values for N, i.e., 11, 13, 15, and 17.

Searching for the optimal (C,γ) pair is also difficult because the data set used here is extremely large. In [50], Lin and Lin found an optimal pair, $(C,\gamma) = (2, 0.125)$, for the PSSP problem with a much smaller data set (about 10 times smaller compared to the data set used here). Despite the difference of data sizes, we find that their optimal pair also benefits our search as a proper starting point. During our search, we change only one parameter at a time. If the change (increase/decrease) leads to a higher accuracy, we continue to do a similar change (increase/decrease) next time; otherwise, we reverse the change (decrease/increase). Both C and γ are tuned with this scheme.

Results

Tables 3, 4, 5, and 6 show the experimental results for various (C,γ) pairs with the window size $N \in \{11, 13, 15, 17\}$, respectively. Here, Q_3 stands for

Table 3. Q2T prediction accuracies of the C-SVC with different (C,γ) values: window size $N = 11$

		Accuracy			
C	γ	Q_3(%)	Q_α(%)	Q_β(%)	Q_c(%)
1	0.02	*73.8*	71.7	54.0	85.5
1	0.04	*73.8*	72.4	53.9	85.1
1.5	0.03	*73.9*	72.6	54.2	84.9
2	0.04	*73.7*	73.1	54.4	84.0
2	0.045	*73.7*	73.3	54.5	83.8
2.5	0.04	*73.6*	73.3	54.8	83.4
2.5	0.045	*73.7*	73.3	55.2	83.4
4	0.04	*73.3*	73.4	55.9	82.0

Table 4. Q2T prediction accuracies of the C-SVC with different (C, γ) values: window size $N = 13$

		Accuracy			
C	γ	$Q_3(\%)$	$Q_\alpha(\%)$	$Q_\beta(\%)$	$Q_c(\%)$
1	0.02	*73.9*	72.3	54.8	84.9
1.5	0.008	*73.6*	71.4	54.3	85.0
1.5	0.02	*73.9*	72.6	54.7	84.8
1.7	0.04	*74.1*	73.6	54.8	83.4
2	0.025	*74.0*	73.0	55.1	84.3
2	0.04	*74.1*	73.9	55.0	83.9
2	0.045	*74.2*	74.1	55.9	83.5
4	0.04	*73.2*	73.9	55.5	81.7

Table 5. Q2T prediction accuracies of the C-SVC with different (C, γ) values: window size $N = 15$

		Accuracy			
C	γ	$Q_3(\%)$	$Q_\alpha(\%)$	$Q_\beta(\%)$	$Q_c(\%)$
2	0.006	*73.4*	70.8	54.2	85.2
2	0.03	*74.1*	73.6	55.6	84.0
2	0.04	*74.2*	73.9	55.7	83.7
2	0.045	*74.0*	73.7	55.4	83.7
2	0.05	*74.0*	73.7	55.4	83.6
2	0.15	*69.0*	63.3	32.7	91.9
2.5	0.02	*74.0*	73.0	55.6	84.0
2.5	0.03	*74.1*	74.0	55.9	83.5
4	0.025	*74.0*	73.8	55.8	83.4

Table 6. Q2T prediction accuracies of the C-SVC with different (C, γ) values: window size $N = 17$

		Accuracy			
C	γ	$Q_3(\%)$	$Q_\alpha(\%)$	$Q_\beta(\%)$	$Q_c(\%)$
1	0.125	*70.0*	63.6	36.0	91.3
2	0.03	*74.1*	73.5	56.2	83.7
2.5	0.001	*71.3*	68.1	52.4	83.5
2.5	0.02	*74.0*	68.1	52.4	83.5
2.5	0.04	*74.0*	75.0	55.8	83.1

the overall accuracy; Q_α, Q_β, and Q_c are the accuracies for α-helix, β-strand, and coil, respectively.

From these tables, we could see that the optimal (C, γ) values for window size $N \in \{11, 13, 15, 17\}$ are $(1.5, 0.03)$, $(2, 0.045)$, $(2, 0.04)$, and $(2, 0.03)$,

Table 7. Q2T prediction accuracies of the multi-class classifier of BSVM with different (C, γ) values: window size $N = 15$

		Accuracy			
C	γ	$Q_3(\%)$	$Q_\alpha(\%)$	$Q_\beta(\%)$	$Q_c(\%)$
2	0.04	*74.18*	73.90	56.39	84.18
2	0.05	*74.02*	73.68	56.09	83.39
2.5	0.03	*74.20*	73.95	56.85	83.22
2.5	0.035	*74.06*	73.93	56.70	82.99
3.0	0.35	*73.77*	73.88	56.55	82.44

respectively. The corresponding Q_3 accuracies achieved are 73.9%, 74.2%, 74.2%, and 74.1%, respectively. A window size of 13 or 15 seems to be the optimal window size that could most efficiently capture the information hidden in the neighboring residues. The best accuracy achieved is 74.2%, with $N = 13$ and $(C, \gamma) = (2, 0.045)$, or $N = 15$ and $(C, \gamma) = (2, 0.04)$.

The original model of SVMs was designed to do binary classification. To deal with multi-class problems, one usually needs to decompose a large classification problem into a number of binary classification problems. The LIBSVM that we used does such a decomposition with the "one-against-one" scheme [59].

In 2001, Crammer and Singer proposed a direct method to build multi-class SVMs [60]. We also applied such a multi-class SVMs to PSSP with the BSVM (http://www.csie.ntu.edu.tw/ cjlin/bsvm/). The results are shown in Table 7. Through comparing Table 5 and Table 7, we found that the multi-class SVMs using Crammer and Singer's scheme [60] and the group of binary SVMs using "one-against-one" scheme [59] obtained similar results.

4.2 T2T Prediction

The T2T prediction uses the output of the Q2T prediction as its input. In T2T prediction, we use the same SVMs as the ones we use in the Q2T prediction. Therefore, we also adopt the same parameter tuning strategy as in the Q2T prediction.

Results

Table 8 shows the best accuracies reached for window size $N \in \{15, 17, 19\}$ with the corresponding C and γ values. From Table 8, it is unexpectedly observed that the structure-structure prediction has actually degraded the prediction performance. A close look at the accuracies for each secondary structure class reveals that the prediction for the coils becomes much less accurate. In comparison to the early results (Tables 3, 4, 5 and 6) in the first

Table 8. The T2T prediction accuracies for window size $N = 15, 17$, and 19

Window Size (N)	C	γ	Accuracy			
			$Q_3(\%)$	$Q_\alpha(\%)$	$Q_\beta(\%)$	$Q_c(\%)$
15	1	2^{-5}	*72.6*	77.9	60.8	74.3
17	1	2^{-4}	*72.6*	78.0	60.4	74.5
19	1	2^{-6}	*72.8*	78.2	60.1	74.9

stage, the Q_c accuracy dropped from 84% to 75%. By sacrificing the accuracy for coils, the predictions for the other two secondary structures improved. However, because coils have a much larger population than the other two kinds of secondary structures, the overall 3-state accuracy Q_3 decreased.

5 Conclusions

To sum up, SVMs performs well in both bioinformatics problems that we discussed in this chapter. For the problem of cancer diagnosis based on microarray data, the SVMs that we used outperformed most of the previously proposed methods in terms of the number of genes required and the accuracy. Therefore, we conclude that the SVMs can not only make highly reliable prediction, but also can reduce redundant genes. For the PSSP problem, the SVMs also obtained results comparable with those obtained by other approaches.

References

1. Cortes C, Vapnik VN (1995) Support vector networks. Machine Learning 20:273–297
2. Vapnik VN (1995) The nature of statistical learning theory. Springer-Verlag, New York
3. Vapnik VN (1998) Statistical learning theory. Wiley, New York
4. Drucker N, Donghui W, Vapnik VN (1999) Support vector machines for spam categorization. IEEE Transaction on Neural Networks 10:1048–1054
5. Chapelle O, Haffner P, Vapnik VN (1999) Support vector machines for histogram-based image classification. IEEE Transaction on Neural Networks 10:1055–1064
6. Hua S, Sun Z (2001) A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. Journal of molecular Biology 308:397–407
7. Strauss DJ, Steidl G (2002) Hybrid wavelet-support vector classification of waveforms. J Comput and Appl 148:375–400
8. Kumar R, Kulkarni A, Jayaraman VK, Kulkarni BD (2004) Symbolization assisted SVM classifier for noisy data. Pattern Recognition Letters 25:495–504

9. Mukkamala S, Sung AH, Abraham A (2004) Intrusion detection using an ensemble of intelligent paradigms. Journal of Network and Computer Applications, In Press
10. Norinder U (2003) Support vector machine models in drug design: applications to drug transport processes and QSAR using simplex optimisations and variable selection. Neurocomputing 55:337–346
11. Van GT, Suykens JAK, Baestaens DE, Lambrechts A, Lanckriet G, Vandaele B, De Moor B, Vandewalle J (2001) Financial time series prediction using least squares support vector machines within the evidence framework. IEEE Transactions on Neural Networks 12:809–821
12. Chang CC, Lin CJ LIBSVM: A library for support vector machines. available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm
13. Schölkopf B, Smolar A, Williamson RC, Bartlett PL (2000) New support vector algorithms. Neural Computation 12:1207–1245
14. Schöklopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC (2001) Estimating the support of a high-dimensional distribution. Neural Computation 13:443–1471
15. Slonim DK (2002) From patterns to pathways: gene expression data analysis comes of age. Nature Genetics Suppl. 32:502–508
16. Russo G, Zegar C, Giordano A (2003) Advantages and limitations of microarray technology in human cancer. Oncogene 22:6497–6507
17. Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A, Boldrick JC, Sabet H, Tran T, Yu X, et al. (2000) Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 403:503–511
18. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA et al. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286:531–537
19. Ma X, Salunga R, Tuggle JT, Gaudet J, Enright E, McQuary P, Payette T, Pistone M, Stecker K, Zhang BM et al. (2003) Gene expression profiles of human breast cancer progression. Proc Natl Acad Sci USA 100:5974–5979
20. Chen X, Cheung ST, So S, Fan ST, Barry C (2002) Gene expression patterns in human liver cancers. Molecular Biology of Cell 13:1929–1939
21. Khan J, Wei JS, Ringner M, Saal LH, Ladanyi M et al. (2001) Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nature Medicine 7:673–679
22. Deutsch JM (2003) Evolutionary algorithms for finding optimal gene sets in microarray prediction. Bioinformatics 19:45–52
23. Tibshirani R, Hastie T, Narashiman B, Chu G (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression. Proc Natl Acad Sci USA 99:6567–6572
24. Bura E, Pfeiffer RM (2003) Graphical methods for class prediction using dimension reduction techniques on DNA microarray data. Bioinformatics 19:1252–1258
25. Troyanskaya O, Cantor M, Sherlock, G et al. (2001) Missing value estimation methods for DNA microarrays. Bioinformatics 17:520–525
26. Dudoit S, Fridlyand J, Speed T (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. J Am Stat Assoc 97:77–87

27. Welch BL (1947) The generalization of student's problem when several different population are involved. Biomethika 34:28–35
28. Tusher, VG, Tibshirani R, Chu G (2001) Significance analysis of microarrays applied to the ionizing radiation response. Proc Natl Acad Sci USA 98:5116–5121
29. Tibshirani R, Hastie T, Narasimhan B, Chu G (2003) Class prediction by nearest shrunken centroids with applications to DNA microarrays. Statistical Science 18:104–117
30. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The protein data bank. Nucleic Acids Research 28:235–242
31. Kendrew JC, Dickerson RE, Strandberg BE, Hart RJ, Davies DR et al. (1960) Structure of myoglobin: a three-dimensional fourier synthesis at 2 Å resolution. Nature 185:422–427
32. Perutz MF, Rossmann MG, Cullis AF, Muirhead G, Will G et al. (1960) Structure of haemoglobin: a three-dimensional fourier synthesis at 5.5 Å resolution. Nature 185:416–422
33. Scheraga HA (1960) Structural studies of ribonuclease III. A model for the secondary and tertiary struture. J Am Chem Soc 82:3847–3852
34. Davids DR (1964) A correlation between amino acid composition and protein structure. Journal of Molecular Biology 9:605–609
35. Robson B, Pain RH (1971) Analysis of the code relating sequence to conformation in proteins: possible implications for the mechanism of formation of helical regions. Journal of Molecular Biology 58:237–259
36. Chou PY, Fasma UD (1974) Prediction of protein conformation. Biochem 13:211–215
37. Lim VI (1974) Structural principles of the globular organization of protein chains. A stereochemical theory of globular protein secondary structure. Journal of Molecular Biology 88:857–872
38. Rost B, Sander C (1994) Combining evolutionary information and neural networks to predict protein secondary structure. Proteins 19:55–72
39. Robson B (1976) Conformational properties of amino acid residues in globular proteins. Journal of Molecular Biology 107:327–56
40. Nagano K (1977) Triplet information in helix prediction applied to the analysis of super-secondary structures. Journal of Molecular Biology 109:251–274
41. Taylor WR, Thornton JM (1983) Prediction of super-secondary structure in proteins. Nature 301:540–542
42. Rooman MJ, Kocher JP, Wodak SJ (1991) Prediction of protein backbone conformation based on seven structure assignments: influence of local interactions. Journal of Molecular Biology 221:961–979
43. Bohr H, Bohr J, Brunak S, Cotterill RMJ, Lautrup B et al (1988) Protein secondary structure and homology by neural networks. FEBS Lett 241:223–228
44. Holley HL, Karplus M (1989) Protein secondary structure prediction with a neural network. Proc Natl Acad Sci USA 86:152–156
45. Stolorz P, Lapedes A, Xia Y (1992) Predicting protein secondary structure using neural net and statistical methods. Journal of Molecular Biology 225:363–377
46. Muggleton S, King RD, Sternberg MJE (1992) Protein secondary structure predictions using logic-based machine learning. Prot Engin 5:647–657

47. Salamov AA, Solovyev VV (1995) Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignment. Journal of Molecular Biology 247:11–15
48. Qian N, Sejnowski TJ (1988) Predicting the secondary structure of globular proteins using neural network models. Journal of Molecular Biology 202:865–84
49. Kneller DG, Cohen FE, Langridge R (1990) Improvements in protein secondary structure prediction by an enhanced neural network. Journal of Molecular Biology 214:171–82
50. Lin KM, Lin CJ (2003) A study on reduced support vector machines. IEEE Transactions on Neural Networks 12:1449–1559
51. Sasagawa F, Tajima K (1993) Prediction of protein secondary structures by a neural network. Computer Applications in the Biosciences 9:147–152
52. Vivarelli F, Giusti G, Villani M, Campanini R, Fraiselli P, Compiani M, Casadio R (1995) LGANN: a parallel system combining a local genetic algorithm and neural networks for the prediction of secondary structure of proteins. Computer Application in the Biosciences 11:763–9
53. Rost B, Sander C (1993) Prediction of protein secondary structure at better than 70% accuracy. Journal of Molecular Biology 232:584–599
54. Rost B (1996) PHD: predicting one-dimensional protein secondary structure by profile-based neural network. Methods in Enzymology 266:525–539
55. Riis SK, Krogh A (1995) Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments. Journal of Computational Biology 3:163–183
56. Jones DT (1999) Protein secondary structure prediction based on position-specific scoring matrices. Journal of Molecular Biology 292:195–202
57. Stephen FA, Warren G, Webb M, Engene WM, David JL (1990) Basic local alignment search tool. Journal of Molecular Biology 215:403–410
58. Altschul SF, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman FJ (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research 25:3389–3402
59. Hsu CW, Lin CJ (2002) A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks 13:415–425
60. Crammer K, Singer Y (2001) On the algorithmic implementation of multiclass kernel-based vector machines. Journal of Machine Learning Research 2:265–292