New feature extraction in gene expression data for tumor classification*

HE Renya, CHENG Qiansheng**, WU Lianwen and YUAN Kehong

(LMAM, School of Mathematical Sciences, Institute of Molecular Medicine, Peking University, Beijing 100871, China)

Received January 31, 2005; revised April 18, 2005

Abstract Using gene expression data to discriminate tumor from the normal ones is a powerful method. However, it is sometimes difficult because the gene expression data are in high dimension and the object number of the data sets is very small. The key technique is to find a new gene expression profiling that can provide understanding and insight into tumor related cellular processes. In this paper, we propose a new feature extraction method based on variance to the center of the class and employ the support vector machine to recognize the gene data either normal or tumor. Two tumor data sets are used to demonstrate the effectiveness of our methods. The results show that the performance has been significantly improved.

Keywords: tumor classification, support vector machine (SVM), bioinformatics, feature extraction, gene expression.

Genome research has become a research area of great interest recently. Different properties of gene expression can be studied using microarray, such as expression at transcription or translation level, and subcellular localization of gene products. By comparing gene expression in normal and disease cells, microarray may be used to identify disease genes and targets for therapeutic drugs, which can help progresses in the prevention and treatment of tumor. Gene expression data are expected to be used in the development of efficient cancer diagnosis and classification platforms[1, 2]. An important aspect of the endeavor is to predict tumor types on the basis of gene expression data. The gene expression data are in high dimension and small size in general, and contain a lot of noises. Classifying those data sets is a hard question in pattern recognition. To discriminate tumor from the normal based on gene expression profiles is a difficult question. The traditional methods, which are concerned with estimating probability density, are not suitable for such problem. Therefore, we should find another way to solve the problems.

The gene expression data analysis for tumor classification includes selecting important genes or deleting the irrelevant genes, and pattern recognition or discriminating tumor from normal.

In this paper, we propose a feature extraction method and use support vector machine (SVM) to deal with the problem. SVM, developed by Vapnik[3], is based on statistical learning theory, and it has been used in a range of pattern recognition problems such as text categorization[4] and face detection[5]. The motivation for the use of SVMs is that DNA microarrary problems can be very high dimensional and have very few training data. SVM suits for this type of situation.

1 Method

One of the major challenges of gene expression data is the long sequence. The array contains a large number of genes (features) and a lot of noises. Meanwhile, the size of training samples which is often only in tens is far smaller than the dimension and conflicts with the high dimension. Thus, feature extraction in this special pattern recognition is more important. Our aim is to develop a new method to find subsets of features, which are the representation of the gene expression data. The idea is helpful to increase the computation efficiency and remove noise from the gene expression data. In addition, limiting the number of features can sometimes reduce the model complexity, avoid any unnecessary computation and thus reduce the risk of over-fitting.

For pattern recognition, in general, we try to estimate a function $f: R^N \to \{1, 2, \cdots, K\}$ using training data.

* Supported by National Natural Science Foundation of China (Grant Nos. 69872003 and 40035010)

** To whom correspondence should be addressed. E-mail: qcheng@pku.edu.cn

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$$(\boldsymbol{X}_1, y_1), \cdots, (\boldsymbol{X}_l, y_l) \in R^N \times \{1, 2, \cdots, K\}, \tag{1}$$

where l is the number of the train samples; K is the number of the classes; $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_l$ are the sample vectors; $y_1, y_2, \cdots, y_l$ are the classes each $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_l$ belongs to.

When $K = 2$, it can be denoted to estimate a function $f: R^N \to \{+1, -1\}$ using the training data.

$$(\boldsymbol{X}_1, y_1), \cdots, (\boldsymbol{X}_l, y_l) \in R^N \times \{+1, -1\}. \tag{2}$$

The new method of feature extraction in this paper is based on the variance to the center of all the classes' centers.

We first define some weights then select feature according to the weights. Details are as follows:

Firstly, we calculate the center vector m_k of each class

$$m_k = \frac{1}{n_k} \sum_{\boldsymbol{X}_i \in k} \boldsymbol{X}_i, \tag{3}$$

where n_k is the number of train samples belonging to the k_{th} class, $\boldsymbol{X}_i$ is a train sample vector belonging to the k_{th} class. We have

$$\sum_{k=1}^{K} n_k = l. \tag{4}$$

Second, we calculate the center vector of the classes' centers

$$\overline{m_k} = \frac{1}{K} \sum_{k=1}^{K} m_k. \tag{5}$$

Thirdly, we calculate the feature variance to the center vector $\overline{m_k}$

$$\sigma_j^2 = \frac{1}{K} \sum_{k=1}^{K} (m_{kj} - \overline{m_{kj}})^2, \quad 1 \leqslant j \leqslant N. \tag{6}$$

When $K = 2$,

$$\sigma_j^2 = (m_{-1,j} - m_{1,j})^2, \quad 1 \leqslant j \leqslant N. \tag{7}$$

Fourthly, we calculate the linear weight of the feature (gene j)

$$q_j = \sigma_j^2 \Big/ \sum_{j=1}^{N} \sigma_j^2, \quad 1 \leqslant j \leqslant N. \tag{8}$$

The larger q_j is, the more important the feature (gene) j will be. We rank the features according to q_j and select the features with larger q_j.

Because the features adopt the mean values corresponding to denoise filter, they are more robust.

2 Support vector machine (SVM)

SVM is a useful tool in pattern recognition. In this section, we briefly introduce SVM. Detailed information can be found in Refs. [3, 6].

For pattern recognition, we try to estimate a function $f: R^N \to \{+1, -1\}$ using training data

$$(\boldsymbol{X}_1, y_1), \cdots, (\boldsymbol{X}_l, y_l) \in R^N \times \{+1, -1\}.$$

2.1 The linear separable case

To design learning algorithms, we propose a class of functions whose capacity can be computed. SV classifiers are based on the class of hyper-planes

$$(\omega \cdot \boldsymbol{X}) - b = 0 \quad \omega \in R^N, \quad b \in R. \tag{9}$$

They are corresponding to the decision functions

$$f(\boldsymbol{X}) = \operatorname{sign}((\omega \cdot \boldsymbol{X}) - b). \tag{10}$$

Rescale ω and b such that the point(s) closest to the hyper-plane satisfies $|(\omega \cdot \boldsymbol{X}_i) - b| = 1$, which implies

$$y_i(\omega \cdot \boldsymbol{X}_i - b) \geqslant 1, \quad i = 1, 2, \cdots, N. \tag{11}$$

The margin measured perpendicularly to the hyper-plane equals $2/\|\omega\|$. To maximize the margin, we thus have to minimize ω subject to (11). We then minimize the function

$$\phi(\omega) = \frac{1}{2} \|\omega\|^2 = \frac{1}{2}(\omega, \omega). \tag{12}$$

The optimal hyper-plane can be uniquely constructed by solving a constrained quadratic programming problem whose solution is $\omega = \sum_i y_i \alpha_i \cdot \boldsymbol{X}_i$ in terms of a subset of training patterns that lies in the margin. These training patterns are called support vectors. The final decision function (12) depends only on the support vectors.

2.2 The linear non-separable case

To construct the optimal hyper-plane in the case when the data are linearly non-separable, we introduce the nonnegative variables $\xi_i \geqslant 0$ and the function

$$\phi(\xi) = (\omega, \omega) + C \sum_i \xi_i. \tag{13}$$

Here, C is a regularization parameter used to decide a tradeoff between the training error and the margin, which we will minimize subject to constraints

$$y_i((\omega \cdot \boldsymbol{X}_i) - b) \geqslant 1 - \xi_i. \tag{14}$$

It can also be solved by quadratic optimization.

Another method is to map the input vector into a very high-dimensional feature space Z through some nonlinear mapping. In this space an optimal separat-

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

ing hyper-plane is constructed only by inner product of two vectors in the feature space $z(\boldsymbol{X}_1)$ and $z(\boldsymbol{X}_2)$

$$(z_i \cdot z) = K(\boldsymbol{X}, \boldsymbol{X}_i). \quad (15)$$

The mapping could be computationally time consuming with the traditional methods. Then it will be possible to construct the solutions, which are equivalent to the optimal hyper-plane in the feature space. In other words, one can construct nonlinear decision functions in the input space

$$f(\boldsymbol{X}) = \mathrm{sign}\left(\sum_i y_i \alpha_i K(\boldsymbol{X} \cdot \boldsymbol{X}_i) - b\right). \quad (16)$$

Four typical kernel functions are shown in Table 1.

For a practical problem, only the kernel function and the regularity parameter C are selected to specify one SVM.

Table 1. Four types of SVM kernel functions

Name	Function
Linear	$K(\boldsymbol{X}, \boldsymbol{Y}) = \boldsymbol{X} \cdot \boldsymbol{Y}$
Poly	$K(\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{X} \cdot \boldsymbol{Y} + 1)^d$
Radical basis function	$K(\boldsymbol{X}, \boldsymbol{Y}) = \exp\left(-\frac{\|\boldsymbol{X} - \boldsymbol{Y}\|^2}{2\sigma^2}\right)$
Perceptrion	$K(\boldsymbol{X}, \boldsymbol{Y}) = \tanh(\gamma \boldsymbol{X} \cdot \boldsymbol{Y} - \delta)$

3 Results and discussion

We employ the method of sequential minimal optimization to train SVM, because it is easy to calculate the whole minimum without any extra matrix storage and without using any numerical optimization steps[7]. We use the code written by Marcelo[8]. Before using SVM, the sample vector $\boldsymbol{X} = (x_1, \cdots, x_N)$ is transformed into $\boldsymbol{X}' = (x_1', \cdots, x_N')$ by the normalization

$$x_j' = (x_j - \min(x_j))/(\max(x_j) - \min(x_j)), \quad 1 \leqslant j \leqslant N, \quad (17)$$

where N is the dimension number. The distance we use here is the Euclidean distance.

The two sets of data used in this experiment are introduced below:

Data set I: Gene expression in 13 normal and 14 breast tumor samples. Each sample is represented by 5776 gene expression data[10].

Data set II: Gene expression in 47 acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML) samples. The leukemia data set is available at http://www.genome.wi.mit.edu/MPR[11]. Each sample is represented by 7129 gene expression data. Among the 47 ALL, there are 38B-cell ALL and 9 T-cell ALL.

Since the number of the sample is very small, we cannot remove a portion of the samples from the train set, and use them for testing. Cross-validation is a common method to get the prediction error in such situation. Every time, we remove a single sample and learn from the others. Each sample is tested exactly once. This is called leave-one-out cross validation.

In SVM, it is flexible to choose different types of kernel functions and tradeoff parameters, which are often decided according to the performance on the train set or user's experience. Thus we chose different kernels for the two datasets. We used linear kernel function in the breast data set (Data set I), while polynomial kernel function in the AML-ALL data set (Data set II). We also did the feature extraction with different sizes on the two data sets. After the feature extraction, we used SVM with the same type of kernels as that in the whole feature set. The classification performance in different number of features is shown in Fig. 1.

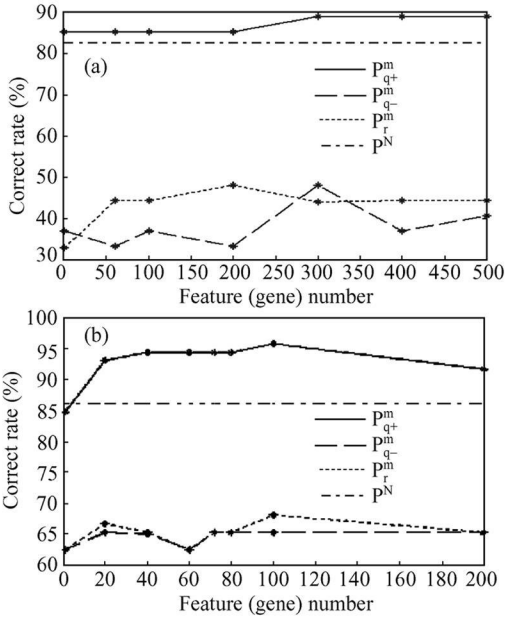

Fig. 1. Test of SVM on (a) the breast data set and (b) the AML-ALL data set using different size of feature set. P^N (the dash-dot line): using all available features; P^m_{q+} (the solid line): using the most important features; P^m_r (the dotted line): using randomly selected features; P^m_{q-} (the dashed line): using the least important features. * represents the real experiment data. The least feature number in the experiments is 1.

The change of the classification performance is

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

plotted with respect to m, the number of retained genes. Also shown is P^N (the dash-dot line), the performance of taking all available features as a reference value. Different (the solid, dotted and dashed) lines denote the performances when using the most important (P^m_{q+}), randomly selected (P^m_r), the least important (P^m_{q-}) features according to (8). In the breast data set, the classification correct rate of the best case is 88.9%, while 95.8% to the AML-ALL data set. In Fig. 1(a) (the breast data set), after feature extraction, the performance can be improved 2.6%—6.3%. In very small feature number such as 60 (That is nearly 1% of the whole gene number), the performance is improved 2.6%. In Fig. 1(b) (the AML-ALL data set), although the SVM using the whole feature performs quite well, we can still get 5.6%—9.7% improvement using a very small part of features. In the very small feature number such as 72 (That is nearly 1% of the whole gene number), the performance is improved 8.3%. Through the experiment, we can see that feature extraction in SVM can improve the classification performance and a very small feature set number can perform well. This is consistent with the case that tumor might occur if only a very small part of genes change a lot suddenly.

In addition, we also use different discriminators to classify data set I in the whole features (genes). The results are shown in Table 2. It can be seen from the results in this table that the SVM is a better performance. Thus, it is correct to use the SVM in this context for validating our new features.

Table 2. Comparison between results by different discriminators of SVM (with Linear Kernel), NN (Nearest Neighborhood) and Fisher using gene expression data set I

	Correct rate (%)	Error rate (%)
SVM	82.6	17.4
NN	81.5	28.5
Fisher	76.2	33.8

4 Conclusions

We propose a feature extraction method and combine it with SVM to get significant results for tumor classification from gene expression data. The more samples are adopted, the better the correct rate of recognition will be. This method is useful to develop the DNA chip and protein chip for clinical application. Because of achievement in human genome, the application of gene expression data attracts more attention. The relative research in methodology is also more important. The results we obtained in this work are only based on limited data, further experiments based on more data are needed. The correct rate is not very high, so there is room for improvement. Our research on feature extraction method will continue. The selection of the kernel, its parameters, such as the tradeoff parameter, and finding the optimal parameter of the Gaussian kernel requires many experiments. In addition, we will make further cooperation with the biologist and clinical doctor to explore the potential of gene expression data in our future research.

Acknowledgements The authors would like to thank Dr. Zhou Xiaobo, Harvard Medical School, for his helpful discussions.

References

1 Dudoit S., Fridlyand J. and Speed T. P. Comparison of discrimination methods for the classification of tumors using gene expression data. UC Berkeley technical report #576, 2000.
2 Wen X. L. Large-scale temporal gene expression mapping of central nervous systems development. PNAS, 95: 334—339.
3 Vapnik V. N. Statistical Learning Theory. New York: John Wiley and Sons., 1998.
4 Joachims T. Text categorization with support vector machines: learning with many relevant features. In: Proc. 10th European CONF. Machine Learning, Springer-Verlag, 1998.
5 Osuna E., Freund R. and Girosi F. Improved training algorithm for support vector machines. In: Proc. IEEE NNSP' 1997, 276—285.
6 Burges C. J. C. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 1998, 2: 121—167.
7 Platt J. C. Fast Training of SVM Using Sequential Minimal Optimization, Advance in Kernel Methods-Support Vector Learning. Cambridge: MIT Press, 1998.
8 Marcelo B. A. Available at http://www.cpdee.ufmg.br/~barros/[2004-12-21].
9 Amir B. D., Laurakey B., Nir F. et al. Tissue classification with gene expression profiles. In: Proceedings of the Fourth Annual International Conference on Computational Molecular Biology (Recomb 2000), Tokyo: Acm Press, Japan.
10 Proceedings of Academy Science, USA, 1999, 96: 9212—9217.
11 Golub T. R. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science, 1999, 286: 531—537.
12 He R. Y. The study of some classification methods on pattern recognition, PhD thesis, Peking University, 2002.

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net