Diagnosing Breast Cancer Based on Support Vector Machines

H. X. Liu,[†] R. S. Zhang,*[,†,‡] F. Luan,[†] X. J. Yao,[†,§] M. C. Liu,[†] Z. D. Hu,[†] and B. T. Fan[§]

Department of Chemistry, Lanzhou University, Lanzhou 730000, China, Department of Computer Science, Lanzhou University, Lanzhou 730000, China, and Université Paris 7-Denis Diderot, ITODYS 1, Rue Guy de la Brosse, 75005 Paris, France

Received November 27, 2002

The Support Vector Machine (SVM) classification algorithm, recently developed from the machine learning community, was used to diagnose breast cancer. At the same time, the SVM was compared to several machine learning techniques currently used in this field. The classification task involves predicting the state of diseases, using data obtained from the UCI machine learning repository. SVM outperformed k-means cluster and two artificial neural networks on the whole. It can be concluded that nine samples could be mislabeled from the comparison of several machine learning techniques.

1. INTRODUCTION

In recent years, neural networks have been successfully applied in function approximation,[1,2] pattern association and pattern recognition,[3−5] etc., concerning various fields concluding mathematics, economics, medicine, chemistry, and many others. Nevertheless, they suffer from many problems and are not well-controlled learning machines. The support vector machine is a new algorithm from the machine learning community. Due to its remarkable generalization performance, the SVM has attracted attention and gained extensive application. For example, for the pattern recognition case, SVMs have been used for isolated handwritten digit recognition,[6] object recognition,[7] face detection in images,[8] text categorization,[9] drug design,[10] prediction of protein structure,[11] and identifying genes, etc.[12]

This article applied SVMs to diagnose breast cancer. Breast cancer, as a kind of cancers, severely threatens feminine health. According to the states of the breast cancer, they can be classified into benign breast tumor and malignant breast tumor (cancer). Aiming at various states of the disease, there will be various Rx and treating medications. Thus, diagnosis is very important.

In this article, several other machine learning techniques were also applied to diagnose this disease in order to identify the reliability of the support vector machines, using data obtained from the UCI machine learning repository.[13]

2. PROBLEM DESCRIPTION

The data used in this experiment were obtained from the UCI machine learning repository[11] and described by Dr. William H. Wolberg. The breast cancer data have been used in some research.[14] We discussed the effect of nine characteristic parameters on the state of breast cancer and the influence of the involved parameter on the performance of the SVM models in this article. At the same time, the comparison between the performance of SVMs and one of other techniques was performed using these data. The problem is to predict the state of breast cancer. In this database, there are 699 pieces of samples, and every sample is expressed by nine characteristic parameters. The nine parameters are as follows: Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, Mitoses. According to the properties of the nine parameters, the breast cancer was classified into benign (expressed by "2") breast tumor and breast cancer (expressed by "4"). Among the 699 samples, there are 16 samples with incomplete parameters. So, the remained 683 samples were used for machine learning. In this article, the data set was randomly divided into two subsets: a training set of 547 samples and a test set of 136 samples.

3. METHODOLOGY

Most of the Classifiers typically learn by empirical risk minimization (ERM) ,[15] that is they search for the hypothesis with the lowest error on the training set. Unfortunately, this approach is doomed to failure without some sort of capacity control.[15,17] To see this, consider a very expressive hypothesis space. If the data are noisy, which is true of most real world applications, then the ERM learner will choose a hypothesis that accurately models the data and the noise. Such a hypothesis will perform badly on unseen data. Note that the SVM is the only algorithm which performs capacity control simultaneously with risk minimization; this is termed structural risk minimization (SRM).

As the theories of neural networks and clustering analysis have been well described in many monographs and articles, we only give a brief description on the simple theory of the SVMs.

3.1. SRM.[15] Suppose we are given 1 observation. Each observation consists of a pair: a vector $\mathbf{x}_i \in R^n$, $i = 1,...,l$ and the associated "truth" $y_i \in \{-1, +1\}$. Now suppose we have a machine whose task is to learn the mapping $\mathbf{x}_i \mapsto y_i$. The machine is actually defined by a set of possible mappings $\mathbf{x} \mapsto f(\mathbf{x},\alpha)$, $\forall\ \mathbf{x},\alpha$. The machine is assumed to be deterministic: for a given input $\mathbf{x}$, and choice of α, it will always give the same output $f(\mathbf{x}_i,\alpha)$. A particular choice of α generates what we will call a "trained machine".

* Corresponding author phone: +86-931-891-2578; fax: +86-931-891-2582; e-mail: ruison@public.lz.gs.cn.
[†] Department of Chemistry, Lanzhou University.
[‡] Department of Computer Science, Lanzhou University.
[§] Université Paris.

10.1021/ci0256438 CCC: $25.00 © 2003 American Chemical Society
Published on Web 04/25/2003

The expectation of the test error for a trained machine is therefore

$$R(\alpha) = \int \frac{1}{2}|y - f(\mathbf{x},\alpha)|dP(\mathbf{x},y) \quad (1)$$

Note that, when a density $p(\mathbf{x},y)$ exists, $dP(\mathbf{x},y)$ may be written as $p(\mathbf{x},y)d\mathbf{x}dy$. This is a nice way of writing the true mean error. The quantity $R(\alpha)$ is called the expected risk or the risk. The "empirical risk" $R_{emp}(\alpha)$ is defined to be just the measured mean error rate on the training set:

$$R_{emp}(\alpha) = \frac{1}{2l}\sum_{i=1}^{l}|y_i - f(\mathbf{x}_i,\alpha)| \quad (2)$$

Note that no probability distribution appears here, $R_{emp}(\alpha)$ is a fixed number for particular choice of αand for a particular training set $\{\mathbf{x}_i,y_i\}$.

The quantity $(1/2)|y_i - f(\mathbf{x}_i,\alpha)|$ is called the loss. Now choose some η such that $0 \le \eta \le 1$. Then for losses taking these values, with probability $1-\eta$, the following bound holds

$$R(\alpha) \le R_{emp}(\alpha) + \sqrt{\left(\frac{h(\log(2l/h) + 1) - \log(\eta/4)}{l}\right)} \quad (3)$$

where h is a non-negative integer called the Vapnik Chervonenkis (VC) dimension and is a measure of the notion of capacity mentioned above. In the following we will call the right-hand side of the equation the "risk bound". The second term on the right-hand side is called the "VC confidence".

The principle of structural risk minimization of SVMs is to minimize the "risk bound" of classifier, namely solving min $[R_{emp}(\alpha) + \sqrt{((h(\log(2l/h)+1)-\log(\eta/4))/(l))}]$. Therefore, SVMs can perform capacity control simultaneously with risk minimization. Then SVMs achieve higher generalization performance than traditional neural networks.

3.2. Support Vector Machines. As a novel type of neural networks, support vector machine (SVM) has gained increasing attention in areas ranging from its original application in pattern recognition to the extended application of regression estimation, due to its remarkable generalization performance. SVM was developed by Vapnik and co-workers in 1995. Based on the Structural Risk Minimization principle which seeks minimize an upper bound of the generalization error rather than minimize the empirical error commonly implemented in other neural networks, SVMs achieve higher generalization performance than traditional neural networks in solving these machine learning problems. Another key property is that unlike the training of other networks, which requires nonlinear optimization with the danger of getting stuck into local minima, training SVMs is equivalent to solving a linearly constrained quadratic programming problem. Consequently, the solution of SVM is always unique and globally optimal.[18]

The SVM method is outlined first for the linearly separable case. Kernel functions are then introduced in order to deal with nonlinear decision surfaces. Finally, for noisy data, when complete separation of the two classes may not be desirable, slack variables are introduced. A complete description to the theory of SVMs for pattern recognition is in tutorials by Osuna et al.[8] and Burges[15] on SVMs.

3.2.1. Linear Decision Surfaces. In this case, there exists an optimal separating hyperplane, whose function is

$$\mathbf{x}_i \cdot \mathbf{w} + b = 0 \quad (4)$$

which implies

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \ge 1, i = 1,...,l \quad (5)$$

By minimizing $1/2\,||\mathbf{w}||^2$ subject to this constraint, the SVM approach tries to find a unique separating hyperplane. Here $||\mathbf{w}||^2$ is the Euclidean norm of $\vec{\mathbf{w}}$, which maximizes the distance between the hyperplane (optimal separating hyperplane) and the nearest data points of each class. The classifier is called the largest margin classifier.

By introducing Lagrange multipliers α_i, the SVM training procedure amounts to solving a convex Quadratic Programming (QP) problem. The solution is a unique globally optimized result, which can be shown to have the following expansion:

$$\mathbf{w} = \sum_{i=1}^{l}\alpha_i y_i \mathbf{x}_i \quad (6)$$

Only if the corresponding $\alpha_i > 0$, *these* $\boldsymbol{x}_i$ are called support vectors.

When an SVM is trained, the decision function can be written as

$$f(\mathbf{x}) = sign(\sum_{i=1}^{l} y_i\alpha_i(\mathbf{x} \cdot \mathbf{x}_i) + b) \quad (7)$$

The solution obtained is often sparse since only those $\mathbf{x}_i$ with nonzero Lagrange multipliers appear in the solution. This is important when the data to be classified are very large, as is often the case in cheminformatics.

3.2.2. Soft Margin Hyperplanes. In the case of noisy data, forcing zero training error will lead to poor generalization. To take into account the fact that some data points may be misclassified, introduce a set of slack variables

$$\xi_i > 0, i = 1,...,l$$

The relaxed separation constraint is given as

$$y_i(w \cdot x_i + b) \ge 1 - \xi_i, i = 1,...,l \quad (8)$$

The optimal separating hyperplane can be found by minimizing

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{l}\xi_i \quad (9)$$

where C is a regularization parameter used to decide a tradeoff between the training error and the margin.

3.2.3. Nonlinear Decision Surfaces. A linear classifier may not be the most suitable hypothesis for the two classes. The SVM can be used to learn nonlinear decision functions by first mapping the data to some higher dimensional feature space and constructing a separating hyperplane in this space, where the mapping is determined by the kernel function. First notice that the only way in which the data appears in the training problem, eq 7, is in the form of dot products, $\mathbf{x} \cdot \mathbf{x}_i$.

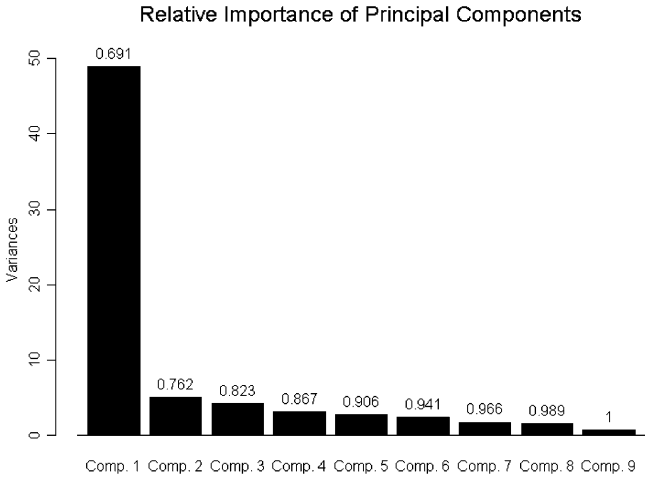

Figure 1. Relative importance of principal components.

Now suppose we first mapped the data to some other (possibly infinite dimensional) Euclidean space H, using a mapping which we will call Φ:

$$\Phi: R^d \mid\rightarrow \mathrm{H} \tag{10}$$

Then, of course, the training algorithm would only depend on the data through dot products in H, i.e., on functions of the form $\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}_i)$. Now if there were a "kernel function" K such that $K(\mathbf{x},\mathbf{x}_i) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}_i)$, we would only need to use K in the training algorithm and would never need to explicitly even know what Φ is.

Thus, the form of the decision function becomes

$$f(\mathbf{x}) = sign(\sum_{i=1}^{l} y_i \alpha_i K(\mathbf{x},\mathbf{x}_i) + b \tag{11}$$

For a given data set, only the kernel function and the regularity parameter *C* must be selected to specify one SVM. For the pattern recognition problem, the first kernels considered are the following:[15]

$K(\mathbf{x},\mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p$ Polynomial kernel function

$K(\mathbf{x},\mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|/2\sigma^2}$ Radial basis kernel function

$K(\mathbf{x},\mathbf{y}) = \tanh(\kappa \mathbf{x} \cdot \mathbf{y} - \delta)$ Sigmoid kernel function

3.3. The Training and Prediction of Breast Cancer Class. Similar with other multivariate statistical models, the performances of SVM classifiers depend on the combination of several parameters. They are capacity parameter *C*, the kernel type K, and its corresponding parameters. *C* is a regularization parameter that controls the tradeoff between maximizing the margin and minimizing the training error. If *C* is too small, then insufficient stress will be placed on fitting the training data. If *C* is too large, then the algorithm will overfit the training data. But, ref 16 indicated that prediction error was scarcely influenced by *C*. To make the learning process stable, a large value should be set up for *C* first (e.g., $C = 100$). The kernel type is another important one. Because the use of SVM models in chemometrics is only in the beginning, there are no clear guidelines on selecting the most effective kernel for a certain classification problem. But for classification tasks, you will most likely use C-classification with the RBF kernel, because of its good general performance and the few number of parameters (only two: *C* and γ).[17]

In this article, the leave-one-out test for the whole training set of 547 samples was carried out to select parameters. The leave-one-out procedure consists of removing one example from the training set, constructing the decision function on the basis only of the remaining training data, and then testing on the removed example. In this fashion one tests all examples of the training data and measures the fraction of errors over the total number of training examples. For the SVM, polynomial kernel functions with powers of 2,3 and radial basis kernel were tested. After building the optimized SVM, the test set was used to predict their class labels. At the same time, the results of k-means cluster and several neural networks were compared with that obtained from SVM.

All calculation programs implementing SVMs were written in R-file based on R1.6.1 software. All neural networks calculations were based on Neural Connection 2.1 package.

4. RESULTS AND DISCUSSION

4.1. Analysis of the Parameters. To identify the importance of the parameters and eliminate redundancy information, the principal components analysis of the original data was done. From Figures 1 and 2, it can be seen that the stronger parameters are as follows: parameter 6 (Bare Nuclei), parameter 2 (Uniformity of Cell Size), parameter 3 (Uniformity of Cell Shape), parameter 8 (Normal Nucleoli), parameter 4 (Marginal Adhesion), and parameter 1 (Clump Thickness), while parameter 5 (Single Epithelial Cell Size), parameter 7 (Bland Chromatin), and parameter 9 (Mitoses) are weaker. To identify the effect of the posterior three parameters further, the models based on the SVM were studied through omitting the three parameters one by one. But for comparison conveniently among different methods and previous work, the model based on nine parameters was built up first.

4.2. The Results from SVMs. 4.2.1. Selection of the Kernel Function and Parameters of the SVM. Given the data set, proper kernel function and its parameters must be chosen to construct the SVM classifier. This selection is important because the type of kernel function determines the sample distribution in the mapping space. There are no successful theoretical methods for determining the optimal kernel function and its parameters. But it is indicated that

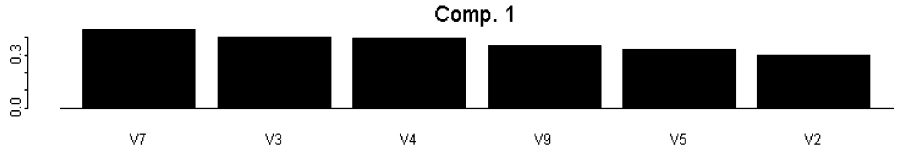

Figure 2. The first component in PCA. Note: V2, V3, V4, V5, V7, V9 are corresponding to the parameters 1, 2, 3, 4, 6, 8, 10.

Table 1. Parameters Selection of the SVM and Corresponding Results

no.	kernel function	*C*	γ	*D*	SV	accuracy	no.	kernel function	*C*	γ	*D*	SV	accuracy
1		100	0.00001		78	96.3437	43		100	0.01	3	49	93.9671
2	radial basis	100	0.00003		62	96.89214	44		100	0.05	3	63	92.68739
3	kernel function	100	0.00005		56	97.07495	45		100	0.1	3	63	92.68739
4		100	0.00007		53	97.07495	46		100	0.5	3	63	92.68739
5		100	0.00009		53	96.89214	47		1	0.0008	3	250	89.76234
6		100	0.0001		53	96.89214	48		10	0.0008	3	110	95.06399
7		100	0.0003		46	96.70932	49		1000	0.0008	3	52	95.97806
8		100	0.0005		46	96.70932	50		2000	0.0008	3	49	95.97806
9		100	0.0007		47	96.5265	51		3000	0.0008	3	53	95.61243
10		100	0.0009		47	96.5265	52		4000	0.0008	3	52	95.2468
11		100	0.001		51	96.5265	53		5000	0.0008	3	53	95.2468
12		100	0.003		57	95.97806	54		6000	0.0008	3	54	95.61243
13		100	0.005		56	95.79525	55		7000	0.0008	3	55	95.61243
14		100	0.007		61	95.06399	56		8000	0.0008	3	55	95.61243
15		100	0.009		59	94.51554	57		9000	0.0008	3	57	95.79525
16		100	0.01		56	94.51554	58		10000	0.0008	3	56	95.61243
17		100	0.05		163	94.88117	59		100	0.0001	2	131	95.42962
18		100	0.1		231	95.2468	60		100	0.0003	2	67	96.70932
19		100	0.5		315	91.5905	61		100	0.0005	2	57	96.89214
20		1	0.00007		229	95.2468	62		100	0.0007	2	54	96.70932
21		10	0.00007		87	96.5265	63		100	0.0009	2	52	96.5265
22		1000	0.00007		48	96.70932	64		100	0.001	2	51	96.3437
23		2000	0.00007		46	96.70932	65		100	0.003	2	48	95.97806
24		3000	0.00007		46	96.70932	66		100	0.005	2	53	94.1499
25		4000	0.00007		46	96.70932	67		100	0.007	2	51	94.1499
26		5000	0.00007		44	96.70932	68		100	0.009	2	51	93.9671
27		6000	0.00007		45	96.70932	69		100	0.01	2	49	93.9671
28		7000	0.00007		46	96.70932	70		100	0.05	2	51	93.05302
29		8000	0.00007		46	96.70932	71		100	0.1	2	48	92.8702
30		9000	0.00007		46	96.70932	72		1	0.0007	2	174	94.51554
31		10000	0.00007		47	96.70932	73		10	0.0007	2	77	96.16088
32		100	0.0001	3	370	75.13711	74		1000	0.0007	2	48	96.16088
33		100	0.0002	3	214	93.05302	75		2000	0.0007	2	48	95.97806
34		100	0.0003	3	141	93.9671	76		3000	0.0007	2	54	95.61243
35		100	0.0004	3	103	95.2468	77		4000	0.0007	2	54	95.06399
36	polynomial	100	0.0005	3	82	95.2468	78		5000	0.0007	2	53	94.51554
37	kernel function	100	0.0006	3	74	95.42962	79		6000	0.0007	2	54	93.9671
38		100	0.0007	3	68	95.97806	80		7000	0.0007	2	52	94.1499
39		100	0.0008	3	66	96.5265	81		8000	0.0007	2	52	94.33272
40		100	0.0009	3	61	96.3437	82		9000	0.0007	2	53	94.1499
41		100	0.001	3	58	96.3437	83		10000	0.0007	2	51	94.1499
42		100	0.005	3	55	94.51554							

the radial basis function and the polynomial function exhibit better performance. So in this article, the radial basis function and the polynomial function were used in the present study. The form of the two kernel functions in R is as follows

Polynomial

$$(\gamma * u' * v + \mathrm{c0})^d$$

Radial basis

$$exp(-\gamma * |u - v|^2)$$

where γ is a constant, the parameter of the two kernels; u and v are two independent variables; and d is the degree of a polynomial function. The calculation results are listed in Table 1.

The statistical results obtained from the SVM experiments are presented in Table 1 and Figures 1 and 2. The calibration results reported in Table 1 are as follows: *C*, capacity parameter, deciding a tradeoff between the training error and the margin; γ, a parameter of kernel function; *d*, degree of the polynomial kernel function; SV, number of support vectors; and accuracy, accuracy of leave-one-out (LOO) cross-validation on the all training samples.

From Table 1 and Figures 1 and 2, it can be found that as a whole the radial basis kernel function performed better than the others and is more appropriate in this context. The first set of models, presented in Table 1 experiments 1−31, was obtained with the radial basis function kernel, researching the effects of fargoing γ and *C* on the accuracy of LOO cross-validation on the all training samples, which results were displayed clearly in Figures 1 and 2. The best results are obtained for $\gamma = 0.00007$, $C = 100$ when the accuracy is up to 97.07495% with 53 support vectors. The second group of models, presented in table experiments 32−83, was obtained with polynomial kernel with degree 2 and degree 3. The best choice for classifying breast cancer is a polynomial of degree 2 and $\gamma = 0.00007$, $C = 100$, when the accuracy is up to 96.70932% with 54 support vectors. This result is worse than that obtained by radial basis function kernel. Therefore, the radial basis kernel function was used to build the followed binary SVMs. After the training, the decision function of the SVM is as follows:

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{53} y_i \alpha_i \exp(-0.00007 * |\mathbf{x} - \mathbf{x}_i|^2) - 0.4957086\right)$$

Table 2. Number of Mistaken Samples of the SVM Method

training set	4,13,100,191,217227,245,252,286,307,339,343, 420,474,475,642
test set	2,50,335

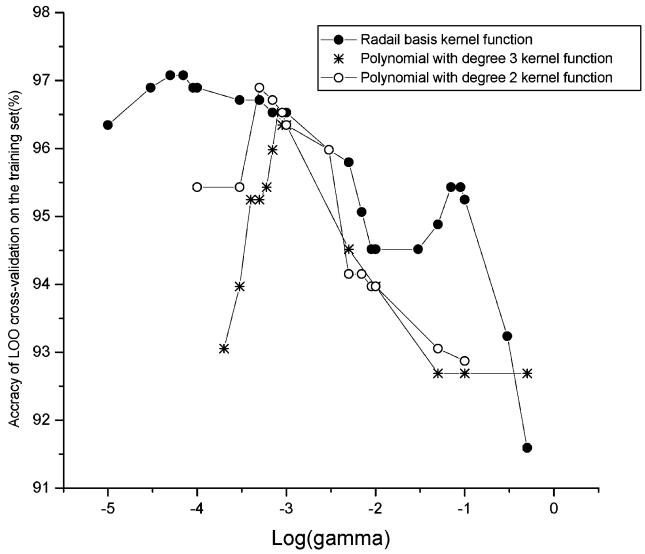

Figure 3. The accuracy of LOO cross-validation of the training set versus log(gamma).

Then, the test set data was tested with the built model. The misclassified samples of the LOO cross-validation on the training set and the test set were listed in the Table 2.

To identify the reliability of the SVM and avoid the effect of randomization, the whole data set was divided into another four groups. Their classified results in optimal condition were listed in Table 3. The comparison between Table 3 and Table 2 displayed that the results from every group are very similar. So it can be concluded that the training samples are representative and there are no effects of randomization.

As can be seen from the experiments 1−19, 32−46, and 59−71 of Table 1 and Figure 3, the performance of the SVMs is very sensitive to γ for selected C, that is, γ greatly affects the learning abilities of machines. For example, to the polynomial kernel with degree 3, among the experiments 32−33, the accuracy varied from 75.13711% to 93.05302% and the number of support vectors varied from 370 to 214 when the γ increased 0.00001. Besides, for the polynomial kernel function, the performance of the SVMs becomes better when γ increases in the beginning and worse as γ continued increasing.

But for the radial basis kernel function, the trend of the performance of the SVMs with γ is more complex. From Figure 1, the best choice for γ to the radial basis kernel and the polynomial kernel function with degree 2,3 is 0.00007, 0.0008, and 0.0007, respectively. For capacity parameter C,

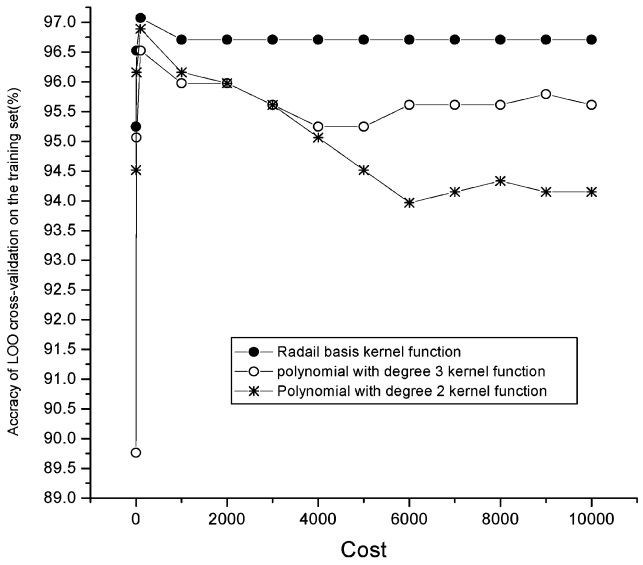

Figure 4. The accuracy of LOO cross-validation of the training set versus C.

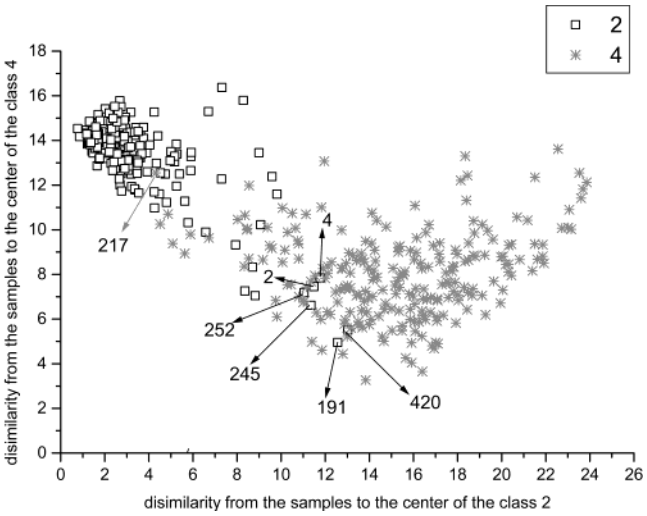

Figure 5. Dissimilarity from the samples to the center of the class 2 versus one of the class 4.

as shown in ref 15, the performance of the SVM becomes better first and then worse, finally insensitive as C increasing from Figure 4.

Besides, because the different type of kernel has different mapping, the kernel type and its corresponding parameters greatly affect the number of support vector. It can be expressed clearly by our experiments. At the same time, the number of support vector has a close relation with the performance of the SVMs and training time. From Table 1, the accuracy is lower when the support vectors are overmuch, which could due to produce overfit. At the same time,

Table 3. Results from Other Groups

group	set	no. of mistaken samples	no. of mistaken samples
group 2	training set	17	2,4,13,50,56,100,191,245,252,286,307,335,339343,420,475,642
	test set	4	58,217,266,441
group 3	training set	17	2,4,13,50,62,100,217,227,245,252,286,339,343,420,441,475,480
	test set	4	191,266,307,335
group 4	training set	12	4,13,100,191,217,227,245,307,339,343,475,480
	test set	5	2,50,252,286,420
group 5	training set	14	2,4,13,100,191,217,227,245,307,335,339,343,420,642
	test set	4	252,286,475,480

Table 4. Results from the SVM through Omitting the Three Parameters One by One

	omitted parameters						
results	Para5	Para7	Para9	Para5 and Para7	Para5 and Para9	Para7 and Para9	Para 5, Para7, and Para9
LOO cross-validation accuracy of training set (%)	96.892	96.344	97.075	96.709	97.075	96.161	96.709
number of mistaken samples of test set	4	3	3	2	3	2	2

Table 5. Initial Centers and Final Centers (Training Set)

		Var1	Var2	Var3	Var4	Var5	Var6	Var7	Var8	Var9
initial centers	cluster1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	cluster2	10.0	10.0	10.0	10.0	10.0	10.0	4.00	10.0	10.0
final centers	cluster1	3.06	1.30	1.43	1.35	2.09	1.32	2.09	1.26	1.11
	cluster2	7.17	6.80	6.73	5.74	5.48	7.93	6.11	6.04	2.57

Table 6. Number of Mistaken Samples of the k-Means Cluster

training set	4, 13, 16, 49, 56, 58, 64, 100, 102, 191, 217, 245, 252, 266, 286, 307, 339, 343, 420, 441, 475
test set	2, 50, 62, 104, 335

overmuch support vectors make the training time longer. In this context, for example, for the experiments 3, 4, we selected $\gamma = 0.00007$ as the optimal value due to the smaller number of the support vectors. For most problems in cheminformatics, the training time is the same important to the performance of a learning machine.

4.2.2. Effects of the Three Parameters on the Models of SVMs. The results from the principal components analysis indicated that the effects of parameter 5 (Single Epithelial Cell Size), parameter 7 (Bland Chromatin), and parameter 9 (Mitoses) on the classified results are weaker. To identify the effect of the three parameters more exactly, the several models were studied through omitting the three parameters one by one in optimal condition, which results can be seen in Table 4.

From the section 4.2.1, the optimal results with nine parameters are that the LOO cross-validation accuracy of training set and numbers of mistaken samples of test set are 97.075% and 3. Comparing with the results from the Table 4, the effect of the parameters 5, 7, and 9 on the models is weak assuredly.

4.3. Comparison to k-Means Cluster. k-Means cluster algorithm is fit for analysis of macroscale data. In this article, there are 683 samples with nine variables. Thus, the algorithm can be used for diagnosing breast cancer. k-Means cluster analysis uses Euclidian distance. Initial cluster centers are chosen in a first pass of the data, then each additional iteration groups observations based on nearest Euclidian distance to the mean of the cluster. Thus cluster centers change at each pass. The process continues until cluster means do not shift more than a given cutoff value or the iteration limit is reached. Here, the clusters are trained using the training data to get their means, and then the testing data are applied. The result is shown in Tables 5 and 6 where Var1 is Clump Thickness; Var2, Uniformity of Cell Size; Var3, Uniformity of Cell Shape; Var4, Marginal Adhesion; Var5, Single Epithelial Cell Size; Var6, Bare Nuclei; Var7, Bland Chromatin; Var8, Normal Nucleoli; Var9, Mitoses.

From Table 6, it can be calculated that the numbers of misclassified samples of the training set and test set are respectively 21 and 5, and the accurate rate is 96.2% and 96.3%. The result indicates that in this study, the performance of k-means cluster algorithm is worse than that of SVM as displayed in the Table 2.

Table 7. Number of Mistaken Samples of Neural Networks

PNN	training set	4, 13, 16, 25, 56, 58, 100, 102, 191, 217, 245, 252, 266, 286, 307, 343, 420, 441, 474, 475
	test set	2, 50, 62, 104
BP	training set	7, 13, 115, 191, 217, 227, 245, 252, 266, 307, 420, 474
	test set	2, 50, 62, 335

Table 8. Comparison between the SVM and the Two Neural Networks

	no. of the mistaken samples of LOO cross-validation on the training set	no. the of mistaken samples of the test set
BP	12	4
PNN	20	4
SVM	16	3

4.4. Comparison to Neural Network Method. In this research, the neural network method was also applied to this problem. The neural network is a very popular algorithm in pattern recognition field from 1980s. Particularly, the BP (Back Propagation), PNN (Probability Neural Network), and RBF (Radial Basis Function) algorithms were widely applied in this field. The BP and PNN were used in our present study. In the BP neural network, there are a large number of controlling parameters, namely, the number of hidden layers, the number of hidden nodes, the learning rate, the momentum term, epochs, transfer functions, and weights initialization methods. The prediction performance is evaluated using the mean squared error based on leave-one-out cross-validation of the training set. The selected parameters are as follows: the number of hidden layers is 1, the number of hidden nodes is 8, the learning rate is 0.2, the momentum term is 0.9, epochs are 209, the hidden nodes use the tanh-sigmoid transfer function, and the output node uses the linear transfer function. To PNN, the choice of the parameters is simpler. Only the spread need be specialized, and here the spread was chosen as 3.0, according to the accuracy of the leave-one-out cross-validation on the training set. The mistaken samples from the two neural networks were displayed in Table 7. Table 8 gives the comparison between the SVM and the two neural networks.

From Table 8, it can be seen that the results of the SVM is better than PNN, worse than BP on the LOO cross-validation of the training set, but for the testing results of the test set, the SVM is the best one. It indicates that the SVM has the better generalization ability. This is because

SVMs implement the structural risk minimization principle which minimizes an upper bound of the generalization error rather than minimizes the training error. This eventually leads to better generalization than neural networks which implement the empirical risk minimization principle. At the same time, the neural network may not converge to global solutions. The gradient descent BP algorithm optimizes the weights in a way that the summed square error is minimized along the steepest slope of the error surface. Global solution is not guaranteed because the algorithm can get stuck in a local minima of the error surface. In the case of SVMs, training SVMs is equivalent to solving a linearly constrained quadratic programming, and the solution of SVMs is unique, optimal, and global. Compared to the BP network, there are fewer free parameters in the SVM. At the same time, as illustrated in the experiment, the performance of SVMs is insensitive to C when a reasonable value is selected for γ. However, for the BP network, there are a large number of controlling parameters which include the number of hidden layers, the number of hidden nodes, the learning rate, the momentum term, epochs, transfer functions, and weights initialization methods. All of them are selected empirically. It is a difficult task to obtain an optimal combination of parameters which produces the best prediction performance.

4.5. Analysis of the Misclassified Samples. From Tables 2, 6, and 7, it can be found that the samples 13, 191, 217, 245, 252, 307, 420 in the training set and the samples 2, 50 in the test set were misclassified by all the methods. The codes of these samples in the original database are 1041806, 1213375, 1226012, 1017023, 242970, 721482, 1293439, 1002945, and 1108449, respectively. It can be presumed that these samples could be mislabeled and need to be determined further. From the parameters and the label, it seems that the bigger the values of the parameters are, the label is inclined to 4 (breast cancer). According to this rule, there exists the great possibility of mislabeling for these mistaken samples from the data.

From the high rates of LOO cross-validation test, it can be supposed that the homogeneous samples should have the high similarity. To analyze the similarity between the samples and validate the mislabeled samples further, dissimilarity analysis was performed. For intuitionistic express of the results, the figures of dissimilarity from the samples to the center of the class 2 versus dissimilarity from the samples to the center of the class 4 were drawn. From the denseness of the homogeneous samples in Figure 5, it can be concluded that the homogeneous samples have the high similarity. From Figure 5, several samples of one class embedded into the dense region of another, which indicates these samples could be probably mislabeled ones. These samples are as follows: 2, 4, 191, 217, 245, 252, 420. They appeared also in the mistaken samples of all the used methods except that the sample 4 was classified accurately by the BP algorithm. This made clear that the samples 2, 191, 217, 245, 252, and 420 were mislabeled in all probability.

5. CONCLUSION

The above results indicate SVM is an effective and accurate method for aiding clinical diagnosis on breast cancer and can be also used to identify mislabeled data. Compared with the other prediction algorithms, the SVM exhibits the better whole performance due to embodying the Structural Risk Minimization principle. It has some advantages over the other techniques of converging to the global optimum and not to a local optimum. Besides, as only support vectors (only a fraction of all data) are used in the generalization process, the SVM adapts particularly to the problem with a great deal of data in cheminformatics. At last, there are fewer free parameters to be adjusted in the SVM. Then the model selecting process is easy to be controlled. Therefore, the SVM is a very promising machine learning technique from any aspects and will gain more extensive application. Through the PCA analysis and the discussion of the parameters in the SVM model, it can be concluded that the parameter 5 (Single Epithelial Cell Size), parameter 7 (Bland Chromatin), and parameter 9 (Mitoses) can be omitted, which reduced the workload of determining the index and economized time. Application of SVM in QSAR is the emphasis of our following work.

ACKNOWLEDGMENT

The authors thank the Association Franco-Chinoise pour la Recherche Scientifique & Technique (AFCRST) for supporting this study (Program PRA SI 00-05). The authors also thank the R Development Core Team for affording the free R1.6.1 software.

REFERENCES AND NOTES

(1) Yeun, Y. S.; Lee, K. H.; Yang Y. S. Function approximations by coupling neural networks and genetic programming trees with oblique decision trees**.** *Artificial Intelligence Eng.* **1999**, *13*(3), 223−239.

(2) Iannella, N.; Back, A. D. A spiking neural network architecture for nonlinear function approximation**.** *Neural Networks* **2001**, *14*(6−7), 933−939.

(3) Tetteh, J.; Suzuki, T.; Metcalfe, E.; Howells, S. Quantitative Structure−Property Relationship for the Estimation of Boiling Point and Flash Point Using a Radial Basis Function Neural Network. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 491−507.

(4) Xiang, Y. H.; Liu, M. C., Zhang, X. Y.; Zhang, R. S.; Hu, Z. D.; Fan, B. T. Quantitative Prediction of Liquid Chromatography Retention of N-Benzylideneanilines Based on Quantum Chemical Parameters and Radial Basis Function Neural Network. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 592−597.

(5) Pomerleau, D. *Neural network perception for mobile robot guidance*; Kluwer Academic Publishing: 1993.

(6) Can, Y.; Jackle, L. D.; Botton, L. et al. *Learning Algorithms for Classification: A Comparison on Handwritten Digit Recognition, Neural Networks: The Statistical Mechanics Perspective*; Oh, J. H., Kwon, C., Cho, S., Eds.; World Scientific: 1995; pp 261−276.

(7) Blanz, V.; Schölkopf, B.; Bülthoff, H.; Burges, C.; Vapnik, V.; Vetter, T. Comparison of view-based object recognition algorithms using realistic 3D models. In *Artificial Neural Networks − ICANN'96*; Malsburg, C. V. D., Seelen, W. V., Vorbrüggen, J. C., Sendhoff, B., Eds.; Springer Lecture Notes in Computer Science, Berlin, 1996; Vol. 1112, pp 251−256.

(8) Osuna, E.; Freund, R.; Girosi, F. Training support vector machines: An application to face detection. In *Proceedings of Computer Vision and Pattern Recognition*; 1997; pp 130−136.

(9) Joachims, T. Text Categorization with Support Vector Machines. LS _ Techical Report, No. 23; University of Dortmund: 1997. ftp://ftpal.informatik.unidortmunddc /pub/report23.ps.Z.

(10) Burbidge, R.; Trotter, M.; Buxton, B.; Holden, S.; Drug design by machine learning: support vector machines for pharmaceutical data analysis. *Comput. Chem.* **2001**, *26*, *5−14*

(11) Cai, Y. D.; Liu, X. J.; Xu, X. B.; Chou, K. C. Prediction of protein structural classes by support vector machines. *Comput. Chem.* **2002**, *26*, 293−296.

(12) Bao, L.; Sun, Z. R. Identifying genes related to drug anticancer mechanisms using support vector machine. *FEBS Lett.* **2002**, *521*, 109−114.
(13) Blake, C. L.; Merz, C. J. UCI Repository of Machine Learning Databases, 1998. URL: http://www.ics.uci.edu/∼mlearn/MLPRepository.html.
(14) Tom D.; Kevin, E. G.; Annette, M.; Extract Simplification of Support Vector Solutions. *J. Machine Learning Res.* **2001**, *2*, 293−297.
(15) Burges, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Mining Knowledge Discovery* **1998**, *2*(2), 1−47.
(16) Wang, W. J.; Xu, Z. B.; Lu, W. Z., Zhang, X. Y. Determination of the spread parameter in the Gaussian kernel for classification and regression. *Neurocomputing* In press.
(17) Bishop, C. *Neural Networks for Pattern Recognition*; Clarendon Press: Oxford, 1997.
(18) Cao, L. J. Support vector machines experts for time series forecasting. *Neurocomputing* **2002**, in press.

CI0256438