Ensemble Classifiers Based on Kernel PCA for Cancer Data Classification

Jin Zhou\textsuperscript{1}, Yuqi Pan\textsuperscript{1}, Yuehui Chen\textsuperscript{1}, and Yang Liu\textsuperscript{2}

\textsuperscript{1}School of Information Science and Engineering, University of Jinan, Jinan 250022, P.R. China
ise_zhouj@ujn.edu.cn
\textsuperscript{2}Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong

Abstract. Now the classification of different tumor types is of great importance in cancer diagnosis and drug discovery. It is more desirable to create an optimal ensemble for data analysis that deals with few samples and large features. In this paper, a new ensemble method for cancer data classification is proposed. The gene expression data is firstly preprocessed for normalization. Kernel Principal Component Analysis (KPCA) is then applied to extract features. Secondly, an intelligent approach is brought forward, which uses Support Vector Machine (SVM) as the base classifier and applied with Binary Particle Swarm Optimization (BPSO) for constructing ensemble classifiers. The leukemia and colon datasets are used for conducting all the experiments. Results show that the proposed method produces a good recognition rate comparing with some other advanced artificial techniques.

Keywords: Cancer data classification, Kernel principal component analysis, Support vector machine, Ensemble classifier, Binary particle swarm optimization.

1 Introduction

The recent advent of DNA microarray technique has made simultaneous monitoring of thousands of gene expressions possible [1]. With this abundance of gene expression data, researchers have started to explore the possibilities of cancer data classification. Quite a number of methods have been proposed in recent years with promising results.

Usually, the classification of gene expression data requires two steps: feature selection and data classification. As the microarray data consists of a few hundreds of samples and thousands or even ten thousands of genes, it is extremely difficult to work in such a high dimension space using traditional classification methods directly. So feature selection methods, which include principal components analysis (PCA), Fisher ratio, t-test, correlation analysis etc, have been proposed and developed to reduce the dimensionality [2]. Along with the feature selection methods, intelligent methods have been applied for microarray data classification, such as artificial neural network (ANN) [3], K nearest neighbor (KNN) [4], decision tree [5] and flexible
neural tree (FNT) [6]. Recent years, ensemble approaches [7] have been put forward. It combines multiple classifiers together as a committee to make more appropriate decisions for classifying microarray data instances. Much research has showed that it can offer improved accuracy and reliability.

In this paper, the gene expression data is firstly preprocessed for normalization in which four steps are taken. Kernel principal component analysis (KPCA) is then applied to extract features. Secondly, an intelligent approach is brought forward, which uses Support Vector Machine (SVM) as the base classifier and applied with Binary Particle Swarm Optimization (BPSO) for constructing ensemble classifiers. The leukemia and colon datasets, which were obtained from the Internet, are used for conducting all the experiments.

The paper is organized as follows: Section 2 introduces the normalization of gene expression data. The feature selection method based on KPCA is described in section 3. The optimal design method for constructing ensemble classifiers is discussed in section 4 and 5. Section 6 gives the experiment results. And in section 7, we present our conclusions.

2 Gene Expression Data Normalization

Due to the noisy nature of dataset provided by microarray experiment, preprocessing is an important step in the analysis of microarray data. The raw intensities have a wide dynamic range. Both datasets have to be normalized to decrease the variation before submitting them to the evolutionary algorithm. In this paper, four steps are taken:

1) If a value is greater than the ceiling 16000 and smaller than the floor 100, this value is replaced by the ceiling/floor.

2) Leaving out the genes with $(\text{max} - \text{min}) \leq 500$, here max and min refer to the maximum and minimum of the expression values of a gene, respectively.

3) Carrying out logarithmic transformation with 2 as the base to all the expression values.

4) For each gene $i$, subtract the mean measurement of the gene $\mu_i$ and divide by the standard deviation $\sigma_i$. After this transformation, the mean of each gene will be zero, and the standard deviation will be one.

3 Feature Selection Based on Kernel PCA

The traditional Principal Component Analysis (PCA) [8] is based exclusively on the second-order statistics with smooth Gaussian distribution. It is difficult to describe the data with non-Gaussian distribution, so the Kernel-based algorithm (KPCA algorithm[9,10]) is proposed for nonlinear PCA. KPCA uses kernel function to obtain the arbitrary high-order correlation between input variants, and find the principal components needed through the inner production between input data.

First of all, a nonlinear mapping $\Phi$ is used to map the input data space $\mathbb{R}^n$ into the feature space $F$:
Φ: \( \mathbb{R}^n \rightarrow \mathbb{F} \)
\( \mathbf{x} \rightarrow \Phi(\mathbf{x}) \) \hspace{1cm} (1)

Correspondingly, a pattern in the original input space \( \mathbb{R}^n \) is mapped into a potentially much higher dimensional feature vector in the feature space \( \mathbb{F} \).

An initial motivation of KPCA is to perform PCA in the feature space \( \mathbb{F} \). Let us construct the covariance matrix in the feature space \( \mathbb{F} \):

\[
\Sigma = \frac{1}{M} \sum_{j=1}^{M} (\Phi(\mathbf{x}_j) - \bar{\Phi})(\Phi(\mathbf{x}_j) - \bar{\Phi})^\top ,
\] \hspace{1cm} (2)

where

\[
\bar{\Phi} = \frac{1}{M} \sum_{j=1}^{M} \Phi(\mathbf{x}_j) .
\] \hspace{1cm} (3)

However, it is not easy to centralize data directly in the feature space \( \mathbb{F} \). To avoid this difficulty, we make the assumption again that

\[
\sum_{j=1}^{M} \Phi(\mathbf{x}_j) = 0 .
\] \hspace{1cm} (4)

So let us consider the following noncentralized covariance matrix:

\[
\tilde{\Sigma} = \frac{1}{M} \sum_{j=1}^{M} \Phi(\mathbf{x}_j)\Phi(\mathbf{x}_j)^\top .
\] \hspace{1cm} (5)

Now we have to solve the Eigenvalue equation:

\[
\lambda \mathbf{V} = \tilde{\Sigma} \mathbf{V} ,
\] \hspace{1cm} (6)

for Eigenvalues \( \lambda \geq 0 \) and Eigenvectors \( \mathbf{V} \in \mathbb{F} \setminus \{0\} \).

It is very computationally intensive or even impossible to calculate \( \tilde{\Sigma} \)'s eigenvectors in a high-dimensional (even infinite-dimensional) feature space. KPCA can be viewed as utilizing two key techniques to solve this problem artfully. One is the SVD technique\[11\] adopted in Eigenfaces, and the other is the so-called kernel-tricks\[9\]. SVD technique can be used to transform the eigenvector calculation problem of a large-size matrix to the eigenvector calculation problem of a small-size matrix and, kernel-tricks can be used to avoid the computation of dot products in the feature space by virtue of the following formula:

\[
K(x_j, x_j) = \langle \Phi(x_j), \Phi(x_j) \rangle .
\] \hspace{1cm} (7)

Specifically, let \( Q = [\Phi(x_1), ..., \Phi(x_M)] \); then \( \tilde{\Sigma} \) can also be expressed by

\[
\tilde{\Sigma} = \frac{1}{M} QQ^\top .
\] \hspace{1cm} (8)

Let us form the matrix \( \tilde{R} = Q^\top Q \) : By virtue of kernel-tricks, we can determine the elements of the \( M \times M \) matrix \( \tilde{R} \) by

\[
\tilde{R}_{ij} = \Phi(x_i)^\top \Phi(x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle = K(x_i, x_j) .
\] \hspace{1cm} (9)
Let us calculate the orthonormal eigenvectors \( u_1, u_2, \ldots, u_m \) of \( \tilde{R} \) corresponding to \( m \) largest eigenvalues \( \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \). Then, by SVD technique, the orthonormal eigenvectors \( w_1, w_2, \ldots, w_m \) of \( \tilde{C} \) corresponding to \( m \) largest eigenvalues \( \lambda_1, \lambda_2, \ldots, \lambda_m \) are

\[
\begin{align*}
    w_j &= \frac{1}{\sqrt{\lambda_j}} Qu_j, \quad j = 1, \ldots, m. 
\end{align*}
\]

(10)

After the projection of the mapped sample \( \Phi(x) \) onto the eigenvector \( w_j \), we can obtain the \( j \)-th feature

\[
    y_j = w_j^T \Phi(x) = \frac{1}{\sqrt{\lambda_j}} u_j^T Q \Phi(x). 
\]

(11)

The resulting features \( y_1, y_2, \ldots, y_m \) form a KPCA-transformed feature vector \( Y=(y_1, y_2, \ldots, y_m)^T \) for sample \( x \).

4 Data Classification Using Support Vector Machine

There are many kinds of methods for microarray data classification. Since Support Vector Machine (SVM) is suitable for data analysis that deals with few samples and large features, in recent years, most researchers applied Support Vector Machine (SVM) as the base classifier to learn microarray datasets and obtained very good results.

4.1 Support Vector Machine

Support Vector Machine (SVM), which was originally introduced by Vapnik and co-workers [12], is now used in many classification problems. SVM builds up a hyperplane as the decision surface in such a way to maximize the margin of separation between positive and negative examples. SVM achieves this by the structural risk minimization principle. The error rate of a learning machine on the test data is bounded by the sum of the training-error rate and the capacity of this machine depends on the Vapnik Chervonenkis (VC) dimension. Given a labeled set of training samples \( (X_i, Y_i) \), \( i=1, \ldots, M \), where \( X_i \in R^N \) and \( Y_i \in \{-1, 1\} \), the discriminant hyperplane is defined by:

\[
    f(X) = \sum_{i=1}^{M} Y_i \alpha_i K(X_i, X) + b
\]

(12)

where \( K(X, X_i) \) is a kernel function and the sign of \( f(X) \) determines the membership of \( X \).

The selection of an appropriate kernel function is very important for SVM. At present, the selection of kernel function in microarray data classification is mostly artificial and unitary. An improvement scheme can combine several kinds of kernel functions to gain a higher performance.

The polynomial kernel function has good global quality and strong extrapolation ability. As a result of a low polynomial exponent, a higher computation speed can be obtained. To the opposite, the Gauss radial basis function is the locally strong kernel
function. Its interpolation ability will be weakened along with the parameter $\sigma$’s growth. Therefore, to get a kernel function that has high learning capability, strong generalization, both good extrapolation and interpolation abilities, we need to design a mixed kernel function that combine several kinds of kernel functions together.

In this paper, $K_{\text{mix}}$ is adopted as the kernel function in SVM.

$$K_{\text{mix}} = \lambda K_{\text{poly}} + (1-\lambda)K_{\text{rbf}}, \quad \lambda \in (0,1).$$

(13)

4.2 Parameter Optimization with PSO

Particle Swarm Optimization (PSO) [13] is one of the evolutionary optimization methods inspired by nature. Since PSO was first introduced by Kennedy and Eberhart(1995), it has been successfully applied to optimize various continuous nonlinear functions. In this paper, PSO is used to optimize parameters of the SVM.

A population of particles is randomly generated initially. Each particle represents a potential solution and has a position represented by a position vector $x_i$. A swarm of particles moves through the problem space with the moving velocity of each particle represented by a velocity vector $v_i$. At each iteration step $t$, each particle keeps track of the best position among all the particles $p_i(t)$ and its own best position $p_i(t)$, a new velocity for particle $i$ is updated by

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot \text{rand}_1 \cdot (p_i(t) - x_i(t)) + c_2 \cdot \text{rand}_2 \cdot (p_g(t) - x_i(t)),$$

(14)

where $c_1$ and $c_2$ are positive constant and $\text{rand}_1$ and $\text{rand}_2$ are uniformly distributed random number in [0, 1]. The term $v_i$ is limited to the range of $\pm v_{\text{max}}$. If the velocity violates this limit, it is set to its proper limit. Changing velocity in this way can enable the particle $i$ to search around its individual best position, $p_i$, and global best position, $p_g$. Based on the updated velocities, each particle changes its position according to the following equation

$$x_i(t+1) = x_i(t) + v_i(t+1).$$

(15)

5 An Ensemble Classifiers Design with Binary PSO

Selecting several classifiers to construct the committee is better than any one [14]. So we should select appropriate classifiers to form the classification committee. In this paper, we introduce the selection method classifiers ensemble using Binary of Particle Swarm Optimization (BPSO) [15].

5.1 Particle Representation

Suppose $N$ base classifiers are generated after trained by the feature subsets. They are expressed as $C_1$, $C_2$, $C_3$, ..., $C_N$. In this new ensemble approach for Cancer Data Classification, $X_i^k$ is the $i$-th particle in swarm at iteration $k$. It is represented by a $N$-dimensional vector which is introduced to denote the $N$ base classifiers and can be defined as $X_i^k = [x_{i1}^k, x_{i2}^k, ..., x_{in}^k]$, where $x_{ij}^k$ is the position of the $i$-th particle with
respect to $j$-th dimension. A binary value of 1 for the $j$-th dimension implies that $C_j$ is selected in the solution and 0 otherwise.

5.2 Initial Population

$pop^k$ is the set of $Popsize$ particles in the swarm at iteration $k$, i.e. $pop^k = [X_1^k, X_2^k, \ldots, X_{Popsize}^k]$. For each dimension of a particle, a binary value of 0 or 1 is assigned according to a probability of $e$. In particular,

$$x_{ij}^k = \begin{cases} 1, & \cup (0,1) > e, \\ 0, & \text{otherwise} \end{cases},$$

where $V_i^k$ is the velocity of particle $i$ at iteration $k$. It can be described as $V_i^k = [v_{i1}^k, v_{i2}^k, \ldots, v_{in}^k]$, $v_{ij}^k$ is the velocity of particle $i$ with respect to $j$-th dimension.

Velocity values are restricted to some minimum and maximum values, namely $V_i^k = [V_{min}, V_{max}]$ where $V_{min} = -V_{max}$. The velocity of particle $i$ in the $j$-th dimension is established by

$$v_{ij}^0 = V_{min} + \cup (0,1) * (V_{max} - V_{min}).$$

This limit enhances the local search exploration of the problem space.

5.3 Fitness Function

In order to measure individuals, the fitness function should be created. We first generate the validation set $V$ and then calculate the error $E_{vi}^k$ of each individual on $V$ at iteration $k$. $f(X_i^k)$ is the fitness of the $i$-th particle at iteration $k$ depicted as follows:

$$f(X_i^k) = \frac{1}{E_{vi}},$$

$$E_{vi}^k = \sum_{j=1}^{N} x_{ij}^k \times classifier_j,$$

where $N$ is the total number of base classifiers, $x_{ij}^k$ is the position of the $i$-th particle with respect to $j$-th dimension at iteration $k$, $classifier_j$ is the error of the $j$-th base classifier on $V$.

5.4 Finding New Solutions

Since the BPSO algorithm is employed in this study, we need to use two useful functions for generating new solutions, namely a limitative function $H$ to force the real values between 0 and 1, and a piecewise linear function $G$ to force velocity values to be inside the maximum and minimum allowable values.

$$G(v_{ij}^k) = \begin{cases} V_{max}, & \text{if } v_{ij}^k > V_{max}, \\ v_{ij}^k, & \text{if } |v_{ij}^k| \leq V_{max}, \\ V_{min}, & \text{if } v_{ij}^k < V_{min}. \end{cases}$$
After applying the piecewise linear function, the following limitative function is used to scale the velocities between 0 and 1, which is then used for converting them to the binary values. That is

\[
H(v_{ij}^k) = \frac{1}{1 + \frac{|V_{\text{max}} - v_{ij}^k|}{|v_{ij}^k - V_{\text{min}}|}}. \tag{21}
\]

So, new solutions are found by updating the velocity and dimension respectively.

First, we compute the change in the velocity \(v_{ij}^k\) such that

\[
v_{ij}^{k-1} = w * v_{ij}^{k-1} + c_1 \times \text{rand} \times (p_{b_{ij}^k}^{k-1} - x_{ij}^{k-1}) + c_2 \times \text{rand} \times (g_{b_{ij}^k}^{k-1} - x_{ij}^{k-1}), \tag{22}
\]

where \(P_{b_{ij}^k}^{k}\) is the best value of the particle \(i\) obtained until iteration \(k\). The best position associated with the best fitness value of the particle \(i\) obtained so far is called particle best and defined as \(P_{b_{ij}^k}^{k} = [p_{b_{1i}^k}, p_{b_{2i}^k}, ..., p_{b_{ni}^k}]\). \(G_{b}^{k}\) is the best position among all particles in the swarm, which is achieved so far and can be expressed as \(G_{b}^{k} = [g_{b_{1i}^k}, g_{b_{2i}^k}, ..., g_{b_{ni}^k}]\). \(c_1\) and \(c_2\) are social and cognitive parameters and \(\text{rand}_1\) and \(\text{rand}_2\) are uniform random numbers between 0 and 1.

Then we update the velocity \(v_{ij}^k\) by using the piecewise linear function such that

\[
v_{ij}^k = G(v_{ij}^{k-1} + \Delta v_{ij}^{k-1}). \tag{23}
\]

Finally we update the dimension \(j\) of the particle \(i\) such that

\[
x_{ij}^k = \begin{cases} 1, & \text{if } \bigcup (0,1) < H(v_{ij}^k) \\ 0, & \text{otherwise} \end{cases} \tag{24}
\]

6 Experiments

We performed extensive experiments on two benchmark cancer datasets, which were obtained from the Internet, namely the Leukemia and Colon database. The Leukemia dataset consists of 72 samples taken from leukemia patients: 25 samples of AML and 47 samples of ALL [16]. A total of 38 out of 72 samples were used as training data and the remaining samples were used as test data. Each sample contained 7129 gene expression levels. The Colon dataset consists of 62 samples of colon epithelial cells taken from colon-cancer patients [16]. Each sample contains 2000 gene expression levels. A total of 31 out of 62 samples were used as training data and the remaining samples were used as test data.

For this experiment, the normalization procedure is firstly used for preprocessing the raw data. Four steps were taken. Then the KPCA is employed, 60 informative features of each sample are extracted and 9 training datasets are chosen for training the 9 base classifiers. SVM is employed to be the base classifier and PSO is used to
### Table 1. Parameters used in this paper

<table>
<thead>
<tr>
<th>Parameters for KPCA</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$K(x_i, x_j)$: kernel function</td>
<td>RBF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters for SVM</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$: kernel function proportion coefficient</td>
<td>0.95</td>
</tr>
<tr>
<td>$K(X, X_i)$: kernel function</td>
<td>$K_{mix}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters for PSO</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$L$: population size</td>
<td>30</td>
</tr>
<tr>
<td>$w$: weight</td>
<td>1.0</td>
</tr>
<tr>
<td>$c_1$, $c_2$: learning factor</td>
<td>2.0</td>
</tr>
<tr>
<td>$X_{up}$: the upper boundary of x</td>
<td>3.0</td>
</tr>
<tr>
<td>$X_{down}$: the lower boundary of x</td>
<td>-3.0</td>
</tr>
<tr>
<td>$V_{max}$: the max velocity</td>
<td>1.8</td>
</tr>
<tr>
<td>rand$_1$, rand$_2$: uniform random number</td>
<td>(0, 1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters for BPSO</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$L$: population size</td>
<td>30</td>
</tr>
<tr>
<td>$w$: weight</td>
<td>1.0</td>
</tr>
<tr>
<td>$c_1$, $c_2$: learning factor</td>
<td>2.0</td>
</tr>
<tr>
<td>$V_{max}$: the max velocity</td>
<td>1</td>
</tr>
<tr>
<td>rand$_1$, rand$_2$: uniform random number</td>
<td>(0, 1)</td>
</tr>
</tbody>
</table>

### Table 2. Comparison of different approaches on Leukemia dataset

<table>
<thead>
<tr>
<th>Author</th>
<th>Classification Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>This paper</td>
<td>97.1~100</td>
</tr>
<tr>
<td>Furey et al. [17]</td>
<td>94.1</td>
</tr>
<tr>
<td>Li et al. [18]</td>
<td>84.6</td>
</tr>
<tr>
<td>Ben-Dor et al. [19]</td>
<td>91.6~95.8</td>
</tr>
<tr>
<td>Nguyen et al. [20]</td>
<td>94.2~96.4</td>
</tr>
<tr>
<td>Zhao et al. [21]</td>
<td>95.8~97.2</td>
</tr>
</tbody>
</table>

### Table 3. Comparison of different approaches on Colon dataset

<table>
<thead>
<tr>
<th>Author</th>
<th>Classification Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>This paper</td>
<td>93.7~99.7</td>
</tr>
<tr>
<td>Furey et al. [17]</td>
<td>90.3</td>
</tr>
<tr>
<td>Li et al. [18]</td>
<td>94.1</td>
</tr>
<tr>
<td>Ben-Dor et al. [19]</td>
<td>72.6~80.6</td>
</tr>
<tr>
<td>Nguyen et al. [20]</td>
<td>87.1~93.5</td>
</tr>
<tr>
<td>Zhao et al. [21]</td>
<td>85.5~93.3</td>
</tr>
</tbody>
</table>
optimize the parameters for each SVM. Then BPSO is applied for selecting appropriate base classifiers to construct the classification committee.

In our experiment, the parameters that used are shown in Table 1. A comparison of different feature extraction methods and different classification methods for leukemia dataset is shown in Table 2, for colon dataset is shown in Table 3.

7 Conclusions

In this paper, a new ensemble of classifiers is proposed for cancer data classification. The leukemia and colon datasets are used for conducting all the experiments. The raw data is first preprocessed for normalization. Gene features are then extracted based on the KPCA, which greatly reduces dimensionality, as well as maintains the informative features. At last the SVM is employed to construct the classifier committee based on BPSO for classification. The experimental results show that the proposed framework is efficient in recognition rate comparing with some other advanced artificial techniques.

Acknowledgments

This research was partially supported by the Natural Science Foundation of China under contract number 60573065, the Key Subject Research Foundation of Shandong Province and the Natural Science Foundation of Shandong Province under contract number Y2007G33.

References