Multiple-kernel SVM based multiple-task oriented data mining system for gene expression data analysis

Zhenyu Chen \(^{a,\ast}\), Jianping Li \(^{b}\), Liwei Wei \(^{b}\), Weixuan Xu \(^{b}\), Yong Shi \(^{c,d}\)

\(^{a}\)Department of Management Science and Engineering, School of Business Administration, Northeastern University, Shenyang 110819, China
\(^{b}\)Institute of Policy & Management, Chinese Academy of Sciences, Beijing 100080, China
\(^{c}\)College of Information Sciences and Technology, University of Nebraska at Omaha, Omaha, NE 68118, USA

\(\ast\) Corresponding author.
E-mail addresses: zchen@mail.neu.edu.cn (Z. Chen), ljpbcasipm.ac.cn (J. Li), liwei@casipm.ac.cn (L. Wei), wxu@casipm.ac.cn (W. Xu), yshi@gucas.ac.cn (Y. Shi).

1. Introduction

DNA microarray technology makes it possible to measure the simultaneous expression levels of thousands of genes in a single experiment and allow us to investigate the biological molecular state of a living cell. There are numerous potential applications for DNA microarray technology. Especially, it provides valuable insights into the molecular characteristics of cancers. Utilizing the gene expression data might be the most direct way to improve the diagnosis accuracy and discover the potential knowledge from gene expression data of cancer tissue. However, those techniques fall short on some critical areas. These included (a) interpretation of the solution and extracted knowledge. (b) Integrating various sources data and incorporating the prior knowledge into the system. (c) Giving a global understanding of biological complex systems by a complete knowledge discovery framework. This paper proposes a multiple-kernel SVM based data mining system. Multiple tasks, including feature selection, data fusion, class prediction, decision rule extraction, associated rule extraction and subclass discovery, are incorporated in an integrated framework. ALL-AML Leukemia dataset is used to demonstrate the performance of this system.

Gene expression profiling using DNA microarray technique has been shown as a promising tool to improve the diagnosis and treatment of cancer. Recently, many computational methods have been used to discover maker genes, make class prediction and class discovery based on gene expression data of cancer tissue. However, those techniques fall short on some critical areas. These included (a) interpretation of the solution and extracted knowledge. (b) Integrating various sources data and incorporating the prior knowledge into the system. (c) Giving a global understanding of biological complex systems by a complete knowledge discovery framework. This paper proposes a multiple-kernel SVM based data mining system. Multiple tasks, including feature selection, data fusion, class prediction, decision rule extraction, associated rule extraction and subclass discovery, are incorporated in an integrated framework. ALL-AML Leukemia dataset is used to demonstrate the performance of this system.

This research has been partially supported by Grants from National Natural Science Foundation of China (#70621001, #70531040).


© 2011 Elsevier Ltd. All rights reserved.
algorithms including SVM-RFE (Duan, Rajapakse, & Wang, 2005; Guyon, Weston, & Barnhill, 2002; Tang, Zhang, & Huang, 2007) and saliency analysis (Cao, Seng, Gu, & Lee, 2003) are widely used. Class discovery usually refers to identifying previously unknown subclasses adopting the unsupervised techniques. Hierarchical clustering is a popular unsupervised tool due to its intuitive appeal and visualization properties (Eisen, Spellman, Brown, & Botstein, 1998). Another family of clustering method is greedy search-based iterative descent clustering, such as self-organizing map (SOM) (Tamayo et al., 1999), K-means clustering (Li, Weinberg, Darden, & Pedersen, 2001) and Bayesian clustering (Roth & Lange, 2004). Clustering techniques usually face some difficulties including how to choose the right number of clusters and how to evaluate the clustering results (Monti, Tamayo, & Mesirov, 2003).

Class prediction refers to the assignment of particular samples to already-defined classes which could reflect current states or future outcomes (Golub et al., 1999). The widely used methods for class prediction include decision tree (Camp & Slattery, 2002), artificial neural network (ANN) (Ando, Suguro, & Kobayashi, 2003; Tan & Pan, 2005; Tung & Quek, 2005), support vector machine (SVM) (Mao, Zhou, & Pi, 2005; Statnikov, Aliferis, & Tsamardinos, 2005) and so on. It is the limitation of above mentioned methods that each of them can only deal with one task: Gene identification, class discovery or class prediction. However, microarray data mining needs dealing with various tasks. It is necessary to incorporate the tasks into an enlarged system (Matthias et al., 2003).

(1) Feature selection can strongly influence the performance of the methods. The most outstanding character of gene expression data is that it contains a large number of gene expression values (several thousands to tens of thousands) and a relatively small number of samples (a few dozen). It brings great challenges for the commonly used knowledge discovery methods. When the number of features far exceed the number of training samples available, most classification and clustering methods, such as decision tree, ANN and K-means, are sensitive to noise and susceptible to overfitting (Guyon et al., 2002; Monti et al., 2003; Radirovac, Chawla, & Dunker, 2004; Tung & Quek, 2005). Therefore, features selection is a necessary prior stage of the gene expression data mining (Ando et al., 2003; Matthias et al., 2003; Tan & Pan, 2005; Tung & Quek, 2005).

Besides, many features selection methods require a pre-filtering stage to refine the raw feature set to achieve a satisfactory performance (Tung & Quek, 2005; Zhu, Ong, & Dash, 2007). Especially for the wrapper-based features selection methods, it is popular to propose a hybrid wrapper and filter framework to improve the performance.

(2) With the continuous emergence of new DNA data and new array technologies, how to integrate various sources data and incorporate the prior knowledge is another challenging problem (Fellenberg, 2003). Different microarray techniques use different mechanism to measure gene expression levels. It makes the gene expression levels reported by different techniques not comparable with each other. Consequently, a challenge is integrating databases to connect this disparate information as well as performing studies to collectively analyze those datasets from diverse sources that have heterogeneous formats. It is a natural way to adapt normalization to make the gene expression values from different data sources conformable each other (Goh & Kasabov, 2003). Another way is to use statistic methods to combine the experimental results from single source data (Flikov & Skiena, 2003; Hwang et al., 2005).

(3) Considering the high cost and long time of experimental research, various tasks in the medical field including the diagnosis on a disease, the outcome prediction and drug discovery depend on the computational methods. It is necessary to design multiple-task oriented knowledge discovery system as a complete scheme.

Currently, the hybridized pipeline is a commonly used way to integrate multiple-tasks by incorporating a sequence of methods (Kim, Zhou, Morse, & Park, 2005; Matthias et al., 2003; Radijovic et al., 2004; Tan & Pan, 2005; Tung & Quek, 2005; Sethi & Leangsuksun, 2006). For each method used in one system, it is independent to set the initialized values, carry out the optimization algorithm and tune the free parameters. Although each method performs well, their integration usually does not. A small computational error may be transmitted to the following step and enlarged. It increases the uncertainty of the data mining system. For example, a filter method is used to identify the relevant genes, SVM is used to make classification and decision tree is used to extract the comprehensible rules. The selected gene subset, that is usually not optimal by the filter method, may take great effect on the performance of SVM and decision tree. Besides, good classification performance of SVM can not be take advantage of by the next step: Rule extraction by decision tree.

In this paper, a multiple-kernel SVM (MK-SVM) is proposed for multi-task oriented microarray data mining. Unlike standard SVM that is usually viewed as a “black-box”, multiple-kernel SVM based gene expression data mining system is applicable to feature selection, data fusion, class prediction, decision rule extraction, associated rule extraction and subclass discovery.

This paper is organized as follows: In Section 2, we give a series of algorithm. In Section 3, we develop a methodology for multi-task oriented gene expression data mining. Section 4 presents the case study on ALL-AML leukemia dataset. Section 5 summarizes the results and draws a general conclusion.

2. Multi-kernel based SVM: Proposed algorithms

2.1. SVM: A brief introduction

We only give a brief introduction of SVM for a typical binary classification problem. The basic SVM concepts can be found in (Chen, Li, & Wei, 2007; Cristianini & Shawe-Taylor, 2000; Vapnik, 1995).

Given a set of data points \( G = \{(x_i, y_i)\}_{i=1}^n, x_i \in \mathbb{R}^m \) and \( y_i \in \{+1, -1\} \). The decision function of SVM is

\[
f(x) = \langle w, \phi(x) \rangle + b,
\]

where \( \phi(x) \) is a mapping of sample \( x \) from the input space to a high-dimensional feature space. \( \langle \cdot, \cdot \rangle \) denotes the dot product in the feature space. The optimal values of \( w \) and \( b \) can be obtained by solving the following regularized optimization problem:

\[
\min_{w, b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i,
\]

s.t. \( y_i \langle w, \phi(x_i) \rangle + b \geq 1 - \xi_i, \quad i = 1, \ldots, n \),

\[
\xi_i \geq 0,
\]

where \( \xi_i \) is the ith slack variable and \( C \) is the regularization parameter.

This problem is computationally solved using the solution of its dual form:

\[
\max \left\{ \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j k(x_i, x_j) \right\},
\]
where $x_i$ is a Lagrange multiplier which corresponds to the sample $x_i, k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ is a kernel function.

The most commonly used kernel function is Gaussian kernel:

$$k(x, y) = \exp \left(-\sigma^2 \|x - y\|^2\right),$$

where $\sigma^2$ is the kernel parameter.

The generalization performance of SVM is mainly affected by the kernel parameters, for example, $\sigma^2$ and the regularization parameter $C$. They have to be set beforehand and tuned carefully.

### 2.2. Design of multiple-kernels

Multiple-kernel learning replaces the commonly used single fixed kernel, say Gaussian kernel, by multi-kernel (Bach, Lanckriet, & Jordan, 2004; Chen, 2008; Chen et al., 2007; Lanckriet, Cristianini, Bartlett, El Ghaoui, & Jordan, 2004; Li, Chen, Wei, & Xu, 2007; Michelielli & Pontil, 2005; Parrado-Hernández, Mora-Jiménez, Arenas-García, Figueiras-Vidal, & Navia-Vázquez, 2003; Sonnenburg, Ratsch, Schafer, & Scholkopf, 2006; Tsang & Kwok, 2006; Wei, Chen, & Li, 2011). Multi-kernel is the ensemble of multiple heterogeneous kernels or parameterizations of a single kernel. It provides the chance to choose an optimal kernel. Moreover, it is also used to improve the transparency of SVM (Chen, 2008; Chen et al., 2007).

**Formulation 1:**

In practice, a multi-kernel can be written as the linear combination of basic kernels:

$$k(x_i, x_j) = \sum_d \beta_d k_d(x_i, x_j), \quad \beta_d \geq 0,$$

where $\beta_d$ is the weight on the corresponding basic kernel $k_d(x_i, x_j).$

Usually, heterogeneous kernels or one kernel being continuously or discretely parameterized are selected as basic kernels.

**Formulation 2:**

A free parameter $w_d$ can be introduced to integrate the expert and situation knowledge:

$$k(x_i, x_j) = \sum_d w_d \beta_d k_d(x_i, x_j), \quad \beta_d \geq 0.$$  

Unlike $\beta_d$ that is learned automatically, $w_d$ is set according to the expert and situation knowledge.

**Formulation 3:**

A multiple-kernel can also be written as the convex combination of basic kernels:

$$k(x_i, x_j) = \sum_d \beta_d k_d(x_i, x_j), \quad \beta_d \geq 0, \quad \sum_d \beta_d = 1.$$  

For the corresponding applications, versatile multiple-kernels can be designed by adopting different basic kernels. Three kind of multiple-kernels based on Eq. (7.1) are proposed:

1. A new kernel: Single feature kernel is introduced in Chen et al. (2007) for feature selection via SVM. In the present paper, it is called $K_1$:

$$K_1(x_i, x_j) = \sum_{d=1}^{m} w_d \beta_d k_d(x_{i,d}, x_{j,d}), \quad \beta_d \geq 0,$$

where $x_{i,d}$ denotes the feature in the $d$th dimension of the $i$th input vector.

In Eq. (8), the parameter $\beta_d$ which is named as feature coefficient, represents the weight on the corresponding single feature basic kernel. By the introduction of feature coefficients $\beta$, the optimal feature subset can be obtained by an easier way than the usually used combinatorial optimization methods (see details in the next two sections). The selected feature subset is ones having nonzero feature coefficients.

2. Let $G(s) = \left\{ \left( x^{(s)^i}, y^{(s)^i} \right) \right\}_{i=1}^s$, where $x^{(s)^i} \in R^n$ and $y^{(s)^i} \in \{+1, -1\}$ come from the $s$th data source. We define the following multi-kernel to integrate multi-source data:

$$K_s(x_i, x_j) = \sum_{i} \eta_i k_s(x^{(s)^i}_i, x^{(s)^i}_j), \quad \eta_i \geq 0.$$  

3. Similarly, a new multi-kernel can be defined to make feature selection and multi-source data fusion simultaneously:

$$K_s(x_i, x_j) = \sum_{i} \beta_d \sum_{i} \eta_i k_s(x^{(s)^i}_i, x^{(s)^i}_j) = \sum_{d} \beta_d \mu_d k_d(x_i, x_j),$$

where $\mu_d = \beta_d \eta_i \geq 0$. Besides, ANOVA technique (Gunn & Kandola, 2002) can be used in multi-kernel to increase the number of basic kernels.

### 2.3. One-step algorithm

When the multi-kernel in Eq. (7.2) is used, Eqs. (4), (5) are written as:

$$\max \left\{ \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \sum_{d} w_d \beta_d k_d(x_i, x_j) \right\},$$

s.t. \begin{align*}
\sum_{i=1}^n \alpha_i &= 0, \quad 0 \leq x_i \leq C, \quad i = 1, \ldots, n, \quad \beta_d \geq 0.
\end{align*}

Naturally, a two-stage iterative procedure is considered to decompose the original problem into two sub-problems: A quadratic programming and a linear programming that can be solved by standard optimizers. The convergence property of the two-stage iterative algorithm is discussed and proved in Gunn and Kandola (2002), Lee, Kim, and Lee (2006).

The two-stage iterative algorithm typically gives convergent solutions in a few steps, and often it is sufficient to take a one step update (Lee et al., 2006). It results in a low computational cost. However, most other algorithms proposed for microarray data mining are not capable of solving the optimization problems which come with large number of basic kernels (no less than the number of genes) to combine within reasonable time.

**Algorithm 1. One-step algorithm**

**Input:** Input data vector $x_i \in R^n$ and output $y_i \in \{+1, -1\}$.

**Step 1:** Initialization: Set the original regularization parameters $C$ and the kernel parameter $\sigma^2$ at random. The original feature coefficient $p^{(0)}_d$ is set to $\{p^{(0)}_d = 1 | d = 1, \ldots, m \}$.

**Step 2:** Solving the Lagrange coefficient $\alpha$: The Lagrange coefficient $\alpha$ is obtained by solving the following quadratic programming in which the original feature coefficient $\beta^{(0)}_d$ is used:

$$\max \left\{ \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \sum_{d} w_d \beta^{(0)}_d k_d(x_i, x_j) \right\},$$

s.t. \begin{align*}
\sum_{i=1}^n \alpha_i &= 0, \quad 0 \leq x_i \leq C, \quad i = 1, \ldots, n.
\end{align*}

**Step 3:** Solve the feature coefficient $\beta$: The feature coefficient $\beta$ is obtained by solving the following dual linear program-
ming (see Appendix for the derivation in details) in which the Lagrange coefficient $z$ solved in the last step is used:

$$\max \sum_{i=1}^{n} u_i,$$

subject to

$$\sum_{i=1}^{n} u_i y_i \left( \sum_{d=1}^{m} \alpha_d y_d k(x_{i,d}, x_{d}) \right) \leq w_d, \quad d = 1, \ldots, m,$$

$$\sum_{i=1}^{n} u_i y_i = 0,$$

$$0 \leq u_i \leq \lambda, \quad i = 1, \ldots, n.$$

(20)

Step 4: Tune the free parameters and go back to step 2 until the output are optimal.

Output: $f(x_i) = \text{sign} \left( \sum_{d=1}^{m} w_d \beta_d \left( \sum_{i=1}^{n} \alpha_d y_d k(x_{i,d}, x_{d}) \right) + b \right)$

Algorithm 1.1. Feature selection algorithm

Input: Input data vector $x_i \in \mathbb{R}^{n}$ and output $y_i \in \{+1, -1\}$.

Step 1: Initialization: Use the single feature kernel $K_1$, then:

$$K_{d}(x_{i}, x_{j}) = k(x_{i}^{(d)}, x_{j}^{(d)}).$$

Step 2: Train an MK-SVM by one step algorithm (Algorithm 1).

Output: $f(x_i) = \text{sign} \left( \sum_{d=1}^{m} w_d \beta_d \left( \sum_{i=1}^{n} \alpha_d y_d k(x_{i,d}, x_{d}) \right) + b \right)$

Algorithm 1.2. Data fusion algorithm

Input: Input data vector $x_i \in \mathbb{R}^{n}$ and output $y_i \in \{+1, -1\}$.

Step 1: Initialization: Use the single feature kernel $K_2$, then:

$$K_{d}(x_{i}, x_{j}) = k(x_{i}^{(d)}, x_{j}^{(d)}).$$

Step 2: Train an MK-SVM by one step algorithm (Algorithm 1).

Output: $f(x_i) = \text{sign} \left( \sum_{d=1}^{m} w_d \beta_d \left( \sum_{i=1}^{n} \alpha_d y_d k(x_{i,d}, x_{d}) \right) + b \right)$

2.4. Direct and hybrid algorithms

A multi-kernel with the constraint $\sum_{d=1}^{m} \beta_d = 1$ is designed:

$$k(x_{i}, x_{j}) = \sum_{d=1}^{m} \beta_d k_d(x_{i}, x_{j}), \quad \beta_d > 0, \quad \sum_{d=1}^{m} \beta_d = 1,$$

Let $\gamma_{i,d} = \alpha_{i,d} \beta_{d}$, then Eqs. (11), (12) change into (see (Chen, 2008) for the derivations in detail):

$$\max \left\{ \sum_{d=1}^{m} \gamma_{i,d} - \frac{1}{2} \sum_{i,j,d} \gamma_{i,d} y_i y_j k_d(x_{i}, x_{j}) \right\},$$

subject to

$$\sum_{d=1}^{M} \sum_{i=1}^{N} y_{i,d} \gamma_{i,d} = 0,$$

$$0 \leq \sum_{d=1}^{M} \gamma_{i,d} \leq C, \quad i = 1, \ldots, n,$$

$$\gamma_{i,d} \geq 0, \quad d = 1, \ldots, M.$$

(22)

In the two-stage iterative algorithm, two hyperparameters ($\alpha, \beta$) need to be optimized iteratively. Unlike it, above formulation uses the multi-kernel with the constraint $\sum_{d=1}^{m} \beta_d = 1$, and result in only one hyperparameter $\gamma$.

It is the advantage that the hyperparameter $\gamma$ can be obtained by any standard optimizer of SVM directly. The details of SVM optimizers can be found in Cristianini and Shawe-Taylor (2000).

Algorithm 2. Direct algorithm

Input: Input data vector $x_i \in \mathbb{R}^{n}$ and output $y_i \in \{+1, -1\}$.

Step 1: Initialization: Set the original regularization parameters $C$ and the kernel parameter (i.e. $\sigma^2$) at random.

Step 2: Train an MK-SVM described in Eq. (22), (23) by a standard optimizer of SVM.

Step 3: Tune the free parameters and go back to step 2 until the output are optimal.

For microarray data, sometimes the coefficients matrix ($n \times m$) is too large to be optimized. In Chen et al. (2007), a 1-norm based linear programming is formulated to further reduce the computational cost. Alternatively, we propose a hybrid algorithm in which one step algorithm (see Algorithm 1) is used firstly to reduce the number of features to some dozens. Then direct algorithm (Algorithm 2) is used to refine the results and deliver the results to Algorithm 5 for rule extraction.

Algorithm 3. Hybrid algorithm

Input: Input data vector $x_i \in \mathbb{R}^{n}$ and output $y_i \in \{+1, -1\}$.

Step 1: Train an MK-SVM by one step algorithm (Algorithm 1) and obtain a set of feature subset.

Step 2: In the selected feature subspace in the last step, train an M-SVM by direct algorithm (Algorithm 2).

2.5. Rule extraction algorithm

Rule Extraction Algorithms include three components: Decision rules extraction, rule reduction and ordering and associate rules extraction (see Algorithm 5).

Algorithm 4. Rule extraction from MK-SVM

Input: Input data vector $x_i \in \mathbb{R}^{n}$ and output $y_i \in \{+1, -1\}$.

Step 1: Do for $t = 1, \ldots T$

(1) Set $w^{(i)}_d$ according to expert knowledge.

(2) Train an MK-SVM by hybrid algorithm (Algorithm 3) and obtain support vectors.

(3) Extract decision rules from MK-SVM (see (Chen, 2008; Chen et al., 2007) in details) and obtain a set of rules.

(4) Refine the extracted rules in the last step by rule reduction and ordering (see (Chen, 2008; Chen et al., 2007) in details).

(5) Discover new subclasses (Algorithm 5).

Step 2: Extract associate rules from MK-SVM (algorithm 6) and obtain a set of associate rule set.

Step 3: Refine the discovered new subclasses.

In Chen et al. (2007), we give the approach of decision rule extraction from linear SVM. For example, the extracted decision rules in the 2-dimensional space are written as follows:

\[ \text{IF } S V_1 \leq x_1 \leq +\infty \text{ AND } -\infty \leq x_2 \leq S V_2, \text{THEN } y = 1(-1), \]

(24)

where $SV$ denotes a support vector.

In Chen (2008), we extend linear-SVM based decision rule extraction approach to the nonlinear case. Moreover, the computational complexities are compared.

MK-SVM, unlike traditional unsupervised learning, gives a new way to discover new subclasses. Two measures are defined for subclasses discovery:

(1) Overlapping: It measures how many times all the premises of two rules are satisfied by one sample and their consequents match the target decision at the same time.

(2) Coverage: It measures how many samples are covered by at least one of pairwise-rule.
The pairwise rules with low overlapping and high coverage rate may reveal the potential subclasses. Thus, in the following algorithm, those rules are viewed as the possible divisions of the subclasses. For class discovery, the validation of the computational results is extremely important. It is accepted that if the discovered subclasses reflect the true structure, then a class predictor based on these subclasses should perform well (Golub et al., 1999).

Algorithm 5. New subclasses discovery from MK-SVM

**Input:** A set of rule obtained by decision rules extraction and refinement algorithms.

**Step 1:** Compute the overlapping and coverage rate of each pairwise-rule.

**Step 2:** Rank all pairwise rules and select one with the lowest overlapping and highest coverage rate.

**Step 3:** Label the samples. The samples covered by one rule are labeled as class 1 and the others as class 2.

**Step 4:** The samples with new labels are used to train a new MK-SVM.

**Step 5:** The other samples, which are not covered by the top one pairwise-rule, are input into the trained MK-SVM to be labeled. MK-SVM can also be used to discover associate rules, and then discover associate genes. The confidence measure is defined as follows:

\[
\text{Confidence} = \frac{\text{the sample size covered by two rules at the same time}}{\text{the total sample size}}
\]

Algorithm 6. Extract associate rules from MK-SVM

**Input:** Two sets of rule

**Step 1:** Compute the confidence measure of two rules in the rule set.

**Step 2:** Rank all pairwise rules and select one with the lowest overlapping and highest coverage rate.

3. Multiple-kernel SVM based gene expression data mining system: A methodology

Fig. 1 shows six applications of MK-SVM to gene expression data mining. The algorithms for those applications have been given in the last section.

MK-SVM based gene expression data mining system is shown in Fig. 2.

4. Experiments

ALL-AML leukemia dataset is used to evaluate the performance of MK-SVM based system. ALL-AML leukemia dataset (Golub et al., 1999) is already divided into the training dataset and the testing dataset. The training dataset consists of 38 bone marrow samples (27 ALL and 11 AML), over 7129 probes from 6817 human genes. Also 34 samples in the testing dataset are provided, with 20 ALL and 14 AML.

ALL-AML leukemia dataset has been analyzed before in various papers. It is highly suited for demonstrating the whole process of MK-SVM based microarray data mining.

In the section, the raw data without any preprocessing are used. The Gaussian kernel is used. The results reported in this section are
obtained on the independent testing set and with a 10-fold cross-validation on the training set. Our implementation is carried out on the Matlab 6.5 development environment.

4.1. Feature selection, data fusion and class prediction

By computational analysis, Guyon et al. reveal that there are significant differences between the distribution of the training and test set in the ALL-AML leukemia dataset (Guyon et al., 2002). They further point out that the training and test set come from different data sources. Therefore, we can conduct the multiple sources data fusion on this dataset.

The initial dataset (38 training and 34 testing) is named as D1. The training and testing set are randomly divided into two equal parts. Then one part of the training set and that of the testing set constitute a new dataset: D2. The other two parts constitute D3.

Firstly, algorithm 2.1 is used to identify optimal diagnostic gene subset. The regularized parameter $\lambda$ in Eq. (20) controls the sparsity of the feature coefficients $\beta$ and the size of selected gene subset. Therefore, it should be tuned carefully (Chen et al., 2007).

When different kernels (Gaussian, $K_1$, $K_2$ and $K_3$), datasets (D1, D2, D3) and kernel parameters ($\sigma^2$) are used, the classification accuracies are shown in Table 1. There are 34 testing samples for D1 with Gaussian and $K_1$ kernels. When super-SVM is trained on D3 (D2, D3) is used as the testing set (36 samples). Note that Gaussian kernel is used as the basic kernel in $K_1$, $K_2$ and $K_3$.

Table 2 shows the selected optimal gene subsets on three training sets.

Since the raw gene expression data contain large variations and high level of noise, SVM with Gaussian kernel don't perform well (see the second column of Table 1). There are three ways to improve its performance:

(1) One way is to choose a suitable preprocessing method to reduce the variations of the data.

(2) Another way is to reduce the feature space via feature selection. Compared with SVM with Gaussian kernel, better classification accuracies are achieved when the single feature kernel ($K_r$) is used. It indicates that an effective feature reduction improves the robustness of SVM on high noise data.

(3) The fusion of multiple source data is the third way. Table 1 show that the classification accuracies of MK-SVM with $K_2$ are far better than those of SVM with Gaussian kernel, and even better than those on reduced feature subset (MK-SVM with $K_r$). It demonstrates that data heterogeneity interferes with the selection of optimal gene subset. MK-SVM with $K_r^2$ achieves the best results. Only one sample in D2 (independent testing set) and three samples in D3 are misclassified.

4.2. Decision rules extraction

Based on the selected gene subsets, Algorithm 4 is used to extract decision rules. The threshold of false-alarm measure is set to 0.1 and the threshold of soundness measure is set to 0.9. The extracted rules on original dataset (38 training and 34 testing) are shown in Table 3. It can be seen that the rule set contains only one marker gene: U46499.

When the thresholds of false-alarm and soundness measure are set to 0.2 and 0.6 respectively, the extracted rules are shown in Table 4. The rule set contains three genes: U46499, M31166 and M77142. In comparison with Table 3, the overall accuracy doesn't change.

The performance of an interpretable model is evaluated by three measures: the number of rules, the number of conditions per rule and the number of samples misclassified. The performance of MK-SVM and some other interpretable models (Fang et al., 2006; He, Tang, Zhang, & Sunderraman, 2006) are shown in Table 5. CART is a kind of classification tree [56]. ANFIS is an adaptive network-based fuzzy inference system (He et al., 2006). FARM-DS is a fuzzy association mining algorithm (He et al., 2006). SVM-RFE (Guyon et al., 2002) is used as the filter to select the related genes for above three rule extraction methods. It can be seen that MK-SVM, which uses small number of rules with few conditions per rule, achieves the highest classification accuracy. That is to say, MK-SVM has

### Table 1

<table>
<thead>
<tr>
<th>$\sigma^2$</th>
<th>Gaussian (D1)</th>
<th>$K_1$ (D1)</th>
<th>Ker$^1$ (D2)</th>
<th>Ker$^2$ (D2)</th>
<th>Ker$^3$ (D2)</th>
<th>Ker$^1$ (D3)</th>
<th>Ker$^2$ (D3)</th>
<th>Ker$^3$ (D3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000</td>
<td>16/34</td>
<td>26/34</td>
<td>32/36</td>
<td>33/36</td>
<td>31/36</td>
<td>26/36</td>
<td>30/36</td>
<td>31/36</td>
</tr>
<tr>
<td>7000</td>
<td>16/34</td>
<td>26/34</td>
<td>32/36</td>
<td>34/36</td>
<td>27/36</td>
<td>30/36</td>
<td>30/36</td>
<td>31/36</td>
</tr>
<tr>
<td>9000</td>
<td>18/34</td>
<td>25/34</td>
<td>33/36</td>
<td>33/36</td>
<td>27/36</td>
<td>31/36</td>
<td>33/36</td>
<td>33/36</td>
</tr>
<tr>
<td>11000</td>
<td>18/34</td>
<td>25/34</td>
<td>34/36</td>
<td>34/36</td>
<td>27/36</td>
<td>31/36</td>
<td>32/36</td>
<td>33/36</td>
</tr>
<tr>
<td>13000</td>
<td>23/34</td>
<td>28/34</td>
<td>35/36</td>
<td>33/36</td>
<td>27/36</td>
<td>31/36</td>
<td>33/36</td>
<td>33/36</td>
</tr>
<tr>
<td>15000</td>
<td>21/34</td>
<td>28/34</td>
<td>35/36</td>
<td>33/36</td>
<td>27/36</td>
<td>31/36</td>
<td>33/36</td>
<td>33/36</td>
</tr>
<tr>
<td>17000</td>
<td>19/34</td>
<td>25/34</td>
<td>35/36</td>
<td>35/36</td>
<td>30/36</td>
<td>31/36</td>
<td>31/36</td>
<td>31/36</td>
</tr>
</tbody>
</table>

### Table 2

Optimal gene subsets on three datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Optimal gene subsets</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>AF009426, D49950, J03930, M31166, M54995, M77142, M81933, U46499, X51804</td>
</tr>
<tr>
<td>D2</td>
<td>L09717, M23197, M81933, M48736, U46499, U57094, X07743, U17838, M31211, U72936, M28170</td>
</tr>
<tr>
<td>D3</td>
<td>AF009426, J03890, M23197, M33195, M63379, M77142, M81933, U46499, X51804, U79288, X65644, X68560, X76648, M83652, M31523, M31523</td>
</tr>
</tbody>
</table>

### Table 3

Decision table 1 (the thresholds of false-alarm and soundness measure are 0.1 and 0.9 respectively).

<table>
<thead>
<tr>
<th>No.</th>
<th>Rule body</th>
<th>Class</th>
<th>False-alarm (on testing set)</th>
<th>Soundness (on testing set)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U46499 ≤ 146</td>
<td>ALL</td>
<td>0/20</td>
<td>18/20</td>
</tr>
<tr>
<td>2</td>
<td>U46499 &gt; 208</td>
<td>AML</td>
<td>0/14</td>
<td>13/14</td>
</tr>
<tr>
<td>Overall</td>
<td></td>
<td></td>
<td>0/34</td>
<td>31/34</td>
</tr>
</tbody>
</table>

### Table 4

Decision Table 2 (the thresholds of false-alarm and soundness measure are 0.2 and 0.6 respectively).

<table>
<thead>
<tr>
<th>No.</th>
<th>Rule body</th>
<th>Class</th>
<th>False-alarm (on testing set)</th>
<th>Soundness (on testing set)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U46499 ≤ 146</td>
<td>ALL</td>
<td>0/20</td>
<td>18/20</td>
</tr>
<tr>
<td>2</td>
<td>M31166 ≤ 174</td>
<td>ALL</td>
<td>0/20</td>
<td>5/20</td>
</tr>
<tr>
<td>3</td>
<td>U46499 &gt; 1049</td>
<td>AML</td>
<td>0/14</td>
<td>11/14</td>
</tr>
<tr>
<td>4</td>
<td>M77142 ≤ 122</td>
<td>AML</td>
<td>2/14</td>
<td>7/14</td>
</tr>
<tr>
<td>Overall</td>
<td></td>
<td></td>
<td>0/34</td>
<td>31/34</td>
</tr>
</tbody>
</table>
good generalization performance and good interpretability simultaneously.

4.3. Associated rule (gene) extraction

Due to the complex interaction of genes, usually there are several groups of genes with similar functions. In Algorithm 4, when a big weight is put on the selected genes, those genes are kept out and the new selected genes may have the same explanation capacity.

When big weights are put on the genes: U46499 and M31166, the identified associated genes are shown in Table 6. Then rules extracted from the first set of associated genes in Table 6 (\( \sigma^2 = 10000 \)) are shown in Table 7.

Table 8 shows the associated rules, associated genes and their confidence measures. There are 25 samples in the testing set covered by two sets of associated rules simultaneously. The relationships (potential pathways during the development of the cancer, for example) between the associated genes: U46499, M31166, M77142 and D49950 should be further revealed by analyzing time series expression data.

This method provides us an opportunity to draw up an efficient treatment plan for individual patient. Three samples listed in Table 9 are used to demonstrate the process of drawing the treatment plan based on two decision tables (Tables 4 and 7). The diagnosis process for these three samples is shown in Table 10. For example, the sample one satisfies the premises of “rule a” and “rule 5” which correspond to the class: ALL leukemia, so the diagnosis result of this sample is ALL leukemia. Thus a drug would be designed to adjust the expression levels of the discovered two targets: U46499 and D49950 to normal ones.

4.4. Subclass discovery

Now we use MK-SVM to discover the subclasses: T-ALL and B-ALL in ALL-AML leukemia dataset. If we set a low threshold of the soundness measure, more rules will be included in the rule set (see Table 11). In those rules, the pairwise rules with low overlapping and high coverage rate measure may reveal the potential subclasses. Then we rank those pairwise rules and select the top one to re-label the ALL samples (see Table 12).

It is an important problem to evaluate and correct the discovered subclasses. The re-labeled samples are used to train a new MK-SVM and make rule extraction, and then the performance of this method is tested on ALL leukemia training dataset (38 samples). The results on the ALL leukemia testing set are omitted here, for only one T-ALL in the testing set. The experimental results are shown in Table 13.

Golub et al. (1999) use SOM to clustering this database. AML samples are separated from ALL samples with four errors. They also separate AML, T-ALL and B-ALL respectively (two errors). In Roth and Lange (2004); Alexandridis, Lin, and Irwin, 2004, Gaussian mixture model that is optimized by expectation–maximization (EM) algorithm is used to make class discovery and class prediction. In Roth and Lange (2004), AML samples are separated from ALL samples with two errors, and T-ALL samples are separated from other samples with one error. The method proposed in

<table>
<thead>
<tr>
<th>Rule no.</th>
<th>Rule body</th>
<th>Class</th>
<th>False-alarm (testing set)</th>
<th>Soundness (testing set)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>D49950 &lt; 54</td>
<td>AML</td>
<td>0/14</td>
<td>8/14</td>
</tr>
<tr>
<td>2</td>
<td>M31166 &gt; 174</td>
<td>ALL</td>
<td>0/20</td>
<td>5/20</td>
</tr>
<tr>
<td>3</td>
<td>M77142 &gt; 122</td>
<td>AML</td>
<td>2/14</td>
<td>7/14</td>
</tr>
<tr>
<td>4</td>
<td>M31166 &lt; 72</td>
<td>AML</td>
<td>3/14</td>
<td>13/14</td>
</tr>
<tr>
<td>5</td>
<td>D49950 &gt; 168</td>
<td>ALL</td>
<td>4/20</td>
<td>9/20</td>
</tr>
<tr>
<td>Overall</td>
<td></td>
<td></td>
<td>2/34</td>
<td>28/34</td>
</tr>
<tr>
<td>Confidence</td>
<td></td>
<td></td>
<td></td>
<td>25/34</td>
</tr>
</tbody>
</table>
Table 11
Decision rule extraction for subclass discovery (47 samples for ALL leukemia).

<table>
<thead>
<tr>
<th>No.</th>
<th>Class</th>
<th>Rule body</th>
<th>False-alarm</th>
<th>Soundness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ALL</td>
<td>L20321 ≥ 45</td>
<td>2/47</td>
<td>21/47</td>
</tr>
<tr>
<td>2</td>
<td>ALL</td>
<td>U72621 ≤ 43</td>
<td>1/47</td>
<td>23/47</td>
</tr>
<tr>
<td>3</td>
<td>ALL</td>
<td>M29064 ≥ −26 and U72621 ≥ 83</td>
<td>2/47</td>
<td>6/47</td>
</tr>
<tr>
<td>4</td>
<td>ALL</td>
<td>U20350 ≥ −29 and Y07759 ≥ 2</td>
<td>0/47</td>
<td>5/47</td>
</tr>
<tr>
<td>5</td>
<td>ALL</td>
<td>U20350 ≥ −56 and U72621 ≥ 24 and L20321 ≤ 44 and M29064 ≥ −87 and M77140 ≥ 317</td>
<td>1/47</td>
<td>9/47</td>
</tr>
<tr>
<td>6</td>
<td>ALL</td>
<td>U20350 ≥ −56 and L20321 ≤ 44 and M77140 ≤ 152 and 29064 ≥ −87 and M20747 ≥ 43</td>
<td>1/47</td>
<td>6/47</td>
</tr>
</tbody>
</table>

Table 12
Overlapping and coverage measures for the pairwise rules.

<table>
<thead>
<tr>
<th>Pairwise rules</th>
<th>1–2</th>
<th>1–3</th>
<th>1–4</th>
<th>1–5</th>
<th>1–6</th>
<th>2–3</th>
<th>2–4</th>
<th>2–5</th>
<th>2–6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overlap</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Coverage</td>
<td>24</td>
<td>24</td>
<td>30</td>
<td>30</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>27</td>
<td>20</td>
</tr>
<tr>
<td>Rank</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

Table 13
Performance of MK-SVM on T-ALL and B-ALL subclasses discovery.

<table>
<thead>
<tr>
<th>No.</th>
<th>Rule body</th>
<th>Class</th>
<th>False-alarm</th>
<th>Soundness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X02160 ≥ 36</td>
<td>T-ALL</td>
<td>0/38</td>
<td>5/38</td>
</tr>
<tr>
<td>2</td>
<td>X58072 ≥ 260</td>
<td>B-ALL</td>
<td>1/38</td>
<td>20/38</td>
</tr>
<tr>
<td>3</td>
<td>Y00264 ≥ 63</td>
<td>B-ALL</td>
<td>0/38</td>
<td>17/38</td>
</tr>
<tr>
<td>4</td>
<td>D13720 ≥ 288</td>
<td>T-ALL</td>
<td>3/38</td>
<td>8/38</td>
</tr>
<tr>
<td>Overall</td>
<td>3/38</td>
<td>38/38</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alexandris et al. (2004) achieves high accuracy with only one error. von Heydebreck, Huber, and Pou斯塔 (2001) use the statistic measures to discover ten highest scoring partitions in which three partitions are similar to the true AML, T-ALL and B-ALL labels. Compared with them, the performance of MK-SVM is promising (one error).

5. Conclusions

This paper proposes a multiple-kernel SVM based gene expression data mining system. Multiple tasks, including feature selection, data fusion, class prediction, decision rule extraction, associated rule extraction and subclass discovery, are incorporated in a system and carried out by multiple-kernel SVM.

Experiments, conducted on ALL-AML leukemia dataset, have indicated that feature selection and data fusion can improve the robustness of SVM on high noise and heterogeneous data. Compared with some other interpretable methods, multiple-kernel SVM, using the compact rule set containing a small number of rules with few conditions per rule (usually no more than three), has achieved high classification accuracy.

References


