

國立交通大學

統計學研究所

碩士論文

非線性基因選取方法

Nonlinear Gene Selection Method



研究生：張淑淨

指導教授：洪慧念 教授

洪志真 教授

中華民國九十三年六月

# 非線性基因選取方法

## Nonlinear Gene Selection Method

研究生：張淑淨

Student : Shu-Jing Chang

指導教授：洪慧念 博士

Advisors : Dr. Hui-Nien Hung

洪志真 博士

Dr. Jyh-Jen Horng Shiau

國立交通大學



Submitted to Institute of Statistics

College of Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master

in

Statistics

June 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

# 非線性基因選取方法

研究生：張淑淨

指導教授：洪慧念 博士

洪志真 博士

國立交通大學統計學研究所

## 摘要

微生物晶片資料通常包含的基因數非常多(數千個)，但相對的腫瘤樣本數不到 100 個。從這些大量的基因中去挑選對於分類具有顯著關係的基因稱為基因選取(*gene or feature selection*)。我們在本文中回顧了一些基因選取的方法以及統計學家對於"大  $p$  小  $n$ "問題的處理。我們著重的方法是 Support Vector Machines (SVMs)，將從模擬實驗去探討線性以及非線性分類問題。對於線性分類問題，我們主要探討基因之間相關性的影響和資料具有部份重疊(overlap)的情況；對於非線性分類問題，我們使用兩種基因選取方法，並比較其重要基因的選取結果及分類精確度。

# Nonlinear Gene Selection Method

Student : Shu-Jing Chang

Advisors : Dr. Hui-Nien Hung

Dr. Jyh-Jen Horng Shiau

Institute of Statistics


National Chiao Tung University

## ABSTRACT

Microarray data contains large number of  $p$  genes (usually several thousands) and small number of  $n$  patients (usually nearly 100 or less). The problem of identifying the features best discriminate among the classes to improve the ability of a classifier is known as *feature selection*. Some current feature selection methods and the problem of dealing with "large  $p$ , small  $n$ " are reviewed. The Support Vector Machines (SVMs) has proofed excellent performance in practice as a classification methodology. For linear classification problem, this paper studies the following two issues: (i) the number of one gene's surrogates somehow affects the importance of the gene; (ii) the case of overlapping classes. For nonlinear classification problem, we utilize two procedures: 1. mapping the original nonlinear separable data to the high dimension space, and then use SVM RFE with linear kernel to find crucial genes; 2. using SVM RFE with nonlinear kernel. Then we compare these two methods on nonlinear toy problem.

## 誌 謝

在統計研究所這兩年的求學日子裡，因為有認真親切的師長、熱心助人的學長姐、互相請益的同學，讓我學習了豐富的統計知識、一些電腦的專業技能、培養了獨立的研究精神，確定了研究方向，完成了此篇論文。我要感謝我的指導教授—洪慧念老師與洪志真老師，他們指引著研究方向並且耐心的指導著我，使我能夠持續不斷的將研究的問題探討的更深入、完整。口試委員—許文郁老師、陳宏老師的悉心指教，使我將論文修正的更清楚、豐富。



在生活上，要感謝家人、朋友的陪伴，讓我在遭遇研究困難時，能使我有放鬆的心情去面對。另外也要感謝我們同一個研究團體的翠英、寶文、巧慧的幫助，不僅帶給我生活上的歡樂，在學業上的互相切磋討論也讓我獲益良多。

最後，將這份小小的成果和大家分享，且希望各位親愛的師長、同學們請多珍重。

張淑淨 謹誌于

國立交通大學統計學研究所

中華民國九十三年六月

## Contents

<b>ABSTRACT (in Chinese)</b>	<b>i</b>
<b>ABSTRACT (in English)</b>	<b>ii</b>
<b>ACKNOWLEDGEMENTS (in Chinese)</b>	<b>iii</b>
<b>CONTENTS</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Literature Review</b>	<b>3</b>
<b>3 Support Vector Machines</b>	<b>5</b>
3.1 Linear Classifier for Linearly Separable Data	6
3.2 Linear Classifier for Overlapping Classes	8
3.3 Nonlinear Classifier for Nonlinearly Separable Data	9
3.4 Popular Kernel and Standard Type of Classification	10
<b>4 Feature Selection Methods for Kernel Machines</b>	<b>11</b>
4.1 Linear Case	11
4.2 Nonlinear Case	12
<b>5 Simulation Studies</b>	<b>13</b>
5.1 Linear problem	13
5.1.1 Overlapping Classes	13
5.1.2 Correlated Data	14
5.2 Nonlinear problem	15
5.2.1 Compare several different cases using nonlinear SVMs	15
5.2.2 Toy experiment	19
<b>6 Conclusion and Future Research</b>	<b>20</b>
<b>Reference</b>	<b>21</b>

# 1 Introduction

Nowadays, the developments of DNA microarrays enable biologists simultaneously to measure thousands of gene expression data and classify samples belonging to different classes. Leukemia dataset containing two types of acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) was originally studied by Golub et al. (1999). Support Vector Machines is a new, powerful, and supervising technology to be used in many real-world applications and proved excellent performance such as the classification problem of microarrays gene expression data (or microarrays data analysis), text categorization, handwritten character recognition, image classification or biological sequence analysis (Markowitz, Edler, and Vingron, 2003). It also successively extended by a number of other researches. The SVM paradigm has a nice geometrical interpretation in the binary case. It creates a maximal margin separating hyperplane between the two classes  $\{+\}$  or  $\{-\}$  from the information of pattern vectors  $\mathbf{x} \in \mathcal{R}^p$ . When the dataset is linearly separable, it is possible to construct the optimal hyperplane. SVMs can also use kernel functions which map original nonlinear separable datasets into a higher dimension feature space to deal with nonlinear classification problem. In this paper, we will discuss two typical problems associated with microarray data analysis:

## (a) Classification analysis

We focus on two-class classification problem. Let the training data set is given as  $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)$ , with  $\mathbf{x}_i \in \mathcal{R}^p, y_i \in \{+1, -1\}$ . The training examples are used to construct a decision function  $d(\mathbf{x})$ . Define a separating hyperplane by  $\{\mathbf{x} : d(\mathbf{x}) = 0\}$ . New observations of the test set are classified according to the decision function:

$$\begin{aligned}
d(\mathbf{x}_r) > 0 &\Rightarrow y_r = +1 \\
d(\mathbf{x}_r) < 0 &\Rightarrow y_r = -1 \\
d(\mathbf{x}_r) = 0 &\Rightarrow \mathbf{x}_r \in \text{decision boundary}
\end{aligned}$$

where  $\mathbf{x}_r$  is an input observation in the test set.

(b) Gene selection (or feature selection)

The problem in gene expression data is that the number of  $p$  genes is very large (usually several thousands) and the number of  $n$  patients is comparatively scarce (usually nearly 100 or less), but many of the genes are irrelevant or even noises to the classification problem. In statistics, this problem is called "curse of dimensionality". For the problem of "large  $p$ , small  $n$ ", the accuracy is very high in assigning the label of the training sample, but very low in the test sample. Guyon et al. (2002) also demonstrate that the feature selected matter more than the classifier used. For the reason of generalization performance of a classifier, economical, and computational considerations, we would like to select a subset of relevant and distinct features which best discriminate among the classes to improve the ability of a classifier. A recently proposed gene selection method, specially tested on microarrays expression data, is called Recursive Feature Elimination (RFE). The idea is using the weights of a classifier to produce a feature ranking.

In this paper, we investigate the Support Vector Machines criteria for feature selection in application to classification problems. This is SVMs based on RFE algorithm (Guyon et al., 2002). For linear classification problem, we are interested in getting more insights on (i) the number of one gene's surrogates somehow affect the importance of the gene; (ii) the case of overlapping classes. For nonlinear classification problem, we use two procedures: one is to map the original nonlinear separable data to the high dimension space, and then use SVM RFE with linear kernel to find crucial genes; another is to use SVM RFE with nonlinear kernel. We compare the classification performance of these two methods on nonlinearly toy problem. The distribution of the simulation data (toy problem) are provided in Weston *et al* (2000). We also discuss the effect of normalization, and which kernel is appropriate for



different data structure and decision rules.

The paper is organized as follows. In Section 2 we review the literature about gene selection and classification problem, in section 3 we describe SVMs and in section 4 we introduce feature selection using SVMs. In Section 5 we present several simulation results for linear and nonlinear classification problems. Finally, section 6 contains the summary of the reviews, conclusions and future research directions.

## 2 Literature Review

In the current (recent) literature, two basic approaches for feature selection are proposed: *filter methods* and *wrapper methods*. The signal-to-noise (S2N) in Golub et al. (1999) is a *filter method*. The correlation coefficients used as ranking criteria is

$$w_i = \frac{\mu_i(+)-\mu_i(-)}{\sigma_i(+)+\sigma_i(-)}, i = 1, \dots, p \quad (1)$$

where  $\mu_i$  and  $\sigma_i$  represents the mean and standard deviation of the gene expression values of gene  $i$  of class (+) or class (-). Furey et al. (2000) used the absolute value of  $w_i$ 's as ranking criterion. Recently, Pavlidis (2000) used

$$\frac{(\mu_i(+)-\mu_i(-))^2}{\sigma_i(+)^2 + \sigma_i(-)^2}, i = 1, \dots, p \quad (2)$$

as ranking criterion, which is similar to Fisher's criterion score. For the perspective of classification, it is important to select distinct but still highly informative features. With the filter method, we may identify a large number of relevant genes, and the identified set likely has heavy redundancy. On the other hand, the selected genes are highly correlated to each other (Krishnapuram, Carin, Hartemink, 2004). Recursive Feature Elimination (RFE), which has been proposed by Guyon et al. (2002) is a *wrapper method*. This method is based on a backward sequential selection in Rakotomamonjy, et al. (2003), starting with all the features, and removing one feature or chunks of features at a time. In Guyon et al. (2002), they also

generalized SVM RFE to nonlinear case. (Fujarewicz, and Wiench, et al. 2003) use recursive feature elimination(RFE), recursive feature replacement(RFR), neighborhood analysis and pure Sebestyen criterion four gene selection methods to find differently expressed genes for the tumor/normal classification of colon tissues, showing that the RFE and RFR methods work much better than other two methods, and from the results of leave-one-out cross-validation (LOOCV), RFR gives better performance for smaller gene subsets; RFE is slightly better for larger gene subsets. For the toy experiment, the datasets were described in Weston et al. (2000). Weston et al. (2000) utilize the toy data to compare the performance of different feature selection methods including standard SVMs, their algorithms and three classical filter methods. Their method is based on finding those features which minimize bounds on the leave-one-out error. This search can be efficiently performed via gradient descent. The three filter methods choose the crucial features based on Pearson correlation coefficients, the Fisher criterion score, and the Kolmogorov-Smirnov test. Grandvalet, and Canu, (2002); Rakotomamonjy, (2003) also compare their feature selection approaches to standard SVMs on these datasets (toy). From these literatures, it is inappropriate to use standard SVMs dealing with nonlinear classification problems. Furthermore, multicategory problems are often regarded as a series of binary problems. Lee et al. (2001) proposed multicategory Support Vector Machines (MSVM), which extend the binary SVM to the multicategory case. Lee & Lee (2002) applied the MSVM to analyze the published multiple cancer types of leukemia dataset in Golub et al. (1999) and small round blue cell tumors (SRBCTs) of childhood data set in Khan et al. (2001). The leukemia data were separated into three classes including AML, ALL B-cell and ALL T-cell; the SRBCTs data were separated into four classes including neuroblastoma(NB), rhabdo-myosarcoma (RMS), non-Hodgkin lymphoma (NHL) and the Ewing family of tumors (EWS).

Next, we review some research works on the role of kernels when using a SVM. It is difficult to find an appropriate kernel function for a particular problem. Many literatures only

report results on the kernel that performed best on their test set. For examples of notably face and pedestrian diction, the results of using Gaussian kernel with appropriate settings of the width parameters  $\sigma$  will be better than any other kernel in Rifkin et al. (2002). From (Guyon, Weston, Barnhill, and Vapnik, 2002), they use the polynomial kernel of degree 2 for the example of XOR problem. Even if a strong theoretical method for selecting a kernel is developed, unless this can be validated using independent test sets on a large number of problems. Methods such as bootstrapping and cross-validation will remain the preferred method for kernel selection, in GUNN et al. (1998).

SVMs involve many hyperparameters including degree  $d$  of a polynomial kernel, Gaussian kernel parameter  $\sigma$ , and penalized parameter  $C$ . It is crucial to choose appropriate values of these parameters to achieve the best generalization performance. The appropriate Kernel parameter implicitly defines the structure of high dimensional feature space where a maximal margin hyperplane will be found. If the kernels are too poor, then the system can not separate the data, Cristianini et al. (1998). Cristianini et al. (1998) presented an algorithm which can automatically learn the kernel parameter. Rakotomamonjy et al. (2003) use nonlinear toy problem to represent the influence of the two parameters  $\sigma$  and  $C$  on the test error, as  $\sigma = 3$ , the best performance for  $C=100$ ; as  $C=100$ , the best performance for  $\sigma = 3$ .

### **3 Support Vector Machines**

This section is an overview of linear and nonlinear classification method called Support Vector Machines. The general classification problem can be considered as the two-class problem. The goal is to separate the two classes from available examples. If the data is linearly separable all the support vectors will lie on the margin and hence the number of support vectors can be very small, in GUNN et al. (1998).

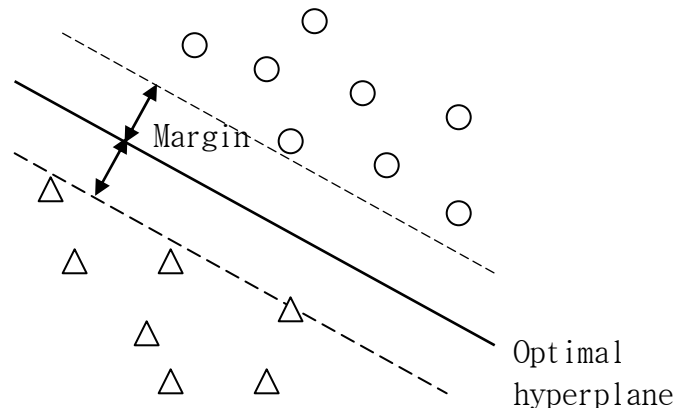


Figure 1: separable case with a maximal margin.

### 3.1 Linear Classifier for Linearly Separable Data

Let the training set is  $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)$ , with  $\mathbf{x}_i \in \mathfrak{R}^p, y_i \in \{+1, -1\}$ . Using the given training examples during the learning stage, the machine finds the parameters  $\mathbf{w} = [w_1 w_2 \dots w_p]^T$  and  $b$  of a decision function  $d(\mathbf{x}, \mathbf{w}, b)$  given as

$$d(\mathbf{x}, \mathbf{w}, b) = \mathbf{w}^T \mathbf{x} + b = \sum_{i=1}^p w_i x_i + b \quad (3)$$

where  $\mathbf{x}, \mathbf{w} \in R^p$ , and the scalar  $b$  is called a *bias*. Define a hyperplane by

$$\{\mathbf{x} : d(\mathbf{x}, \mathbf{w}, b) = 0\} \quad (4)$$

If  $d(\mathbf{x}_r, \mathbf{w}, b) > 0$ , then pattern  $\mathbf{x}_r$  belongs to class 1 (*i.e.*,  $y_r = +1$ )

If  $d(\mathbf{x}_r, \mathbf{w}, b) < 0$ , then pattern  $\mathbf{x}_r$  belongs to class 2 (*i.e.*,  $y_r = -1$ )

The optimal hyperplane is found based on creating the biggest margin between the training points for class 1 and class -1, see Figure 1. In order to find the optimal separating hyperplane, a learning machine should

$$\text{Minimize } \|\mathbf{w}\|$$

$$\text{subject to } y_i[\mathbf{w}^T \mathbf{x}_i + b] \geq 1, \quad i = 1, \dots, n \quad (5)$$

Such an optimization problem is solved by the Lagrange function

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^n \alpha_i \{y_i[\mathbf{w}^T \mathbf{x}_i + b] - 1\}, \quad (6)$$

where the  $\alpha_i$  are Lagrange multipliers.

The corresponding dual objective function is

$$L_d(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j. \quad (7)$$

And the optimal hyperplane is found by maximizing

$$L_d(\alpha) \text{ subject to } \sum_{i=1}^n \alpha_i y_i = 0, \quad \alpha_i \geq 0, i = 1, \dots, n \quad (8)$$

Solutions  $\alpha_i^*$  of this dual optimization problem are easy to be solved, and we can determine the parameters  $\mathbf{w}$  and  $b$  of the optimal hyperplane as follows.

$$\begin{aligned} \mathbf{w}^* &= \sum_{i=1}^n \alpha_i^* y_i \mathbf{x}_i, \quad i = 1, \dots, n \\ b^* &= \frac{1}{N_{SV}} \left( \sum_{s=1}^{N_{SV}} \left( \frac{1}{y_s} - \mathbf{x}_s^T \mathbf{w}^* \right) \right), \quad s = 1, \dots, N_{SV} \end{aligned} \quad (9)$$

where  $N_{SV}$  denotes the number of support vectors.

Finally, we substitute parameters (9) into (3) to obtain an optimal hyperplane  $d^*(\mathbf{x})$  and an indicator function  $i_f$ :

$$\begin{aligned} d^*(\mathbf{x}) &= \mathbf{w}^{*T} \mathbf{x} + b^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}^T \mathbf{x}_i + b^* \\ i_f(\mathbf{x}) &= \text{sign}(d^*(\mathbf{x})). \end{aligned} \quad (10)$$

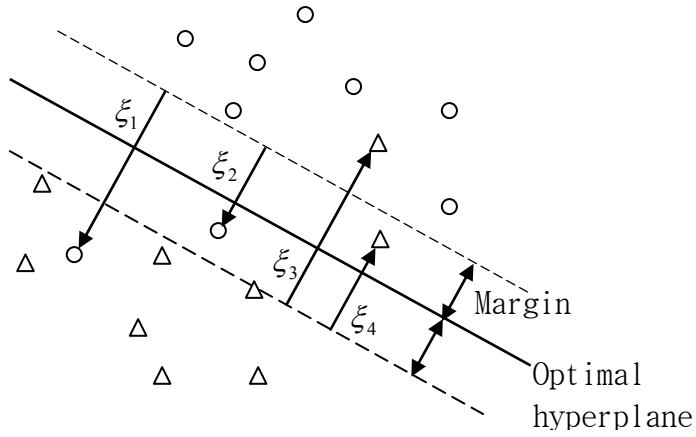


Figure 2: soft decision boundaries with data overlapping

### 3.2 Linear Classifier for Overlapping Classes

In practice, the datasets we want to work are not necessary linearly separable. When the training data sets are inseparable, we can still find a classifier with a maximal margin, allowing some data on the ‘wrong’ side of a decision boundary. See Figure 2, the points labeled  $\xi_i$  are on the wrong side. Then the problem becomes

$$\text{minimize } \|\mathbf{w}\|$$

$$\text{subject to } y_i[\mathbf{w}^T \mathbf{x}_i + b] \geq 1 - \xi_i, i = 1, \dots, n \text{ and,}$$

$$\xi_i \geq 0, \sum \xi_i \leq \text{constat.} \quad (11)$$

Note that  $\sum \xi_i$  is the total proportional amount by which predictions fall on the wrong side of their margin.

For convenient computation, we re-express the problem as

$$\min_{\mathbf{w}, b} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \left( \sum_{i=1}^n \xi_i \right)$$

$$\text{subject to } y_i[\mathbf{w}^T \mathbf{x}_i + b] \geq 1 - \xi_i, \xi_i \geq 0, \quad i = 1, \dots, n \quad (12)$$

where  $C$  is a given constant.

The corresponding dual problem is as follows:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ \text{subject to} \quad & \sum_{i=1}^n \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, \quad i = 1, \dots, n. \end{aligned} \quad (13)$$

where  $C$  is the penalty parameter, determined by the user. Solutions  $\alpha_i^*$  of this dual optimization problem are easy to be solved, and we can determine the parameters  $\mathbf{w}$  and  $b$  of the optimal hyperplane as (9). Finally, we obtain an optimal hyperplane  $d^*(\mathbf{x})$  and an indicator function  $i_f$  as (10). Note that the linearly separable case without data overlapping, this upper bound  $C = \infty$ .

### 3.3 Nonlinear Classifier for Nonlinearly Separable Data

The best way to understand SVMs is using the example of linear decision rule (Ben-dor et al., 2000). In subsection 3.1 and 3.2, we describe optimal hyperplane classifiers and compute linear boundaries in the input feature space. One can make the method more flexible by enlarging the feature space using some basic transformation. Let  $F$  be the enlarged feature space. The idea is to map inputs vectors  $\mathbf{x} \in \mathfrak{R}^p$  into vectors  $\mathbf{z} \in \mathfrak{R}^f = F$ :

$$\mathbf{x} \in \mathfrak{R}^p \rightarrow \mathbf{z}(\mathbf{x}) = [\mathbf{z}_1(\mathbf{x}), \mathbf{z}_2(\mathbf{x}), \dots, \mathbf{z}_f(\mathbf{x})]^T \in \mathfrak{R}^f \quad (14)$$

By performing such a mapping, two classes are easier to be separated by the optimal separating plane in the enlarged feature space  $F$ . A linear boundary in the enlarged feature space  $F$  corresponds to a nonlinear boundary in the original space  $X$ .

We linearly separate images of  $\mathbf{x}$  by applying the linear SVM formulation. The linear classifier (indicator function)  $i_f(\mathbf{x}) = \text{sign}(\mathbf{w}^{*T} \mathbf{z}(\mathbf{x}) + b^*)$  in a feature space  $F$  will create a nonlinear separating hyperplane in the original input space given by

$$i_f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^n \alpha_i^* y_i \langle \mathbf{z}(\mathbf{x}_i), \mathbf{z}(\mathbf{x}) \rangle + b^*\right). \quad (15)$$

Boser et al. (1992) observed that it is not necessary to know the feature space  $F$  explicitly but to calculate the inner products between support vectors of the feature space  $F$  for constructing optimal hyperplane. On the other hand, we need not specify the transformation  $\mathbf{z}(\mathbf{x})$  at all, but require only knowledge of the kernel function  $K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{z}(\mathbf{x}), \mathbf{z}(\mathbf{x}') \rangle$ . Thus replacing  $\langle \mathbf{z}(\mathbf{x}_i), \mathbf{z}(\mathbf{x}) \rangle$  by  $K(\mathbf{x}, \mathbf{x}_i)$  in (14), the separating rule becomes:

$$i_f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^n \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^*\right). \quad (16)$$

where  $\alpha_i^*$  is the solution to the optimization problem

$$\begin{aligned} & \max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ & \text{subject to } \sum_{i=1}^n \alpha_i y_i = 0, \quad C \geq \alpha_i \geq 0, i = 1, \dots, n. \end{aligned} \quad (17)$$

where  $C$  is the margin parameter. When the training data sets are linearly separable for just a few features, the classification is rather insensitive to the value of  $C$ . For convenience, we let  $C=100$  in the linear case study.

### 3.4 Popular Kernel and Standard Type of Classification

We can map the pattern vectors  $\mathbf{x} \in \mathfrak{R}^p$  to a high dimension space  $H$ , and separate there by using a linear kernel function. Given a mapping  $\Phi : \mathfrak{R}^p \rightarrow H$  from input space  $\mathfrak{R}^p$  to an feature space  $H$ , the function  $K : \mathfrak{R}^p \times \mathfrak{R}^p \rightarrow \mathfrak{R}$  is called a kernel function, that is for all  $\mathbf{x}, \mathbf{z} \in \mathfrak{R}^p$ ,  $K(\mathbf{x}, \mathbf{z}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle_H$ . The most commonly used kernels are as follows:

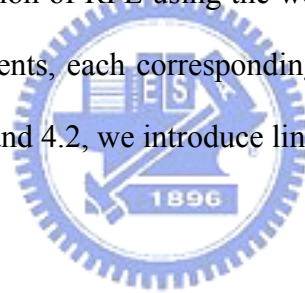


Kernel Functions	Type of Classifier
$K(\mathbf{x}, \mathbf{x}_i) = [(\mathbf{x}^T \mathbf{x}_i) + 1]^d$	polynomial of degree $d$
$K(\mathbf{x}, \mathbf{x}_i) = \exp\left[-\frac{(\mathbf{x} - \mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i)}{2\sigma^2}\right]$	Gaussian RBF (Radial basis function) where $\sigma$ is the width parameter
$K(\mathbf{x}, \mathbf{x}_i) = \tanh[(\mathbf{x}^T \mathbf{x}_i) + b]^*$	Multilayer perceptron

\* Only for certain values of  $b$

## 4 Feature Selection Methods for Kernel Machines

SVM RFE is an application of RFE using the weight magnitude  $\mathbf{w}$  as ranking criterion.  $\mathbf{w}$  is a vector with  $p$  components, each corresponding to expression of a particular gene as ranking criterion. In sec. 4.1 and 4.2, we introduce linear and nonlinear case, respectively.



### 4.1 Linear Case

When our training dataset are linear separable (linear classification problem) then using a linear SVM, SVM RFE use the weight magnitude as ranking criterion. The idea is, starting with all the available genes, build an optimal SVM model, and remove the feature whose associated weight is smallest in absolute value. Repeat this criterion from the surviving genes until only the desired numbers of features remain (Zhang, and Wong, 2001). The criterion is using the following iterative procedure (Guyon et al., 2002):

1. Train the classifier with SVM (optimize the weights  $w_i$ ).
2. Compute the weights for all features.
3. Remove the feature with the smallest absolute weight.

## 4.2 Nonlinear Case

When data is nonlinear separable (nonlinear classification problem), we can use the following two procedures.

### (i) nonlinear SVMs

The basic idea is to remove those features that affect the margin the least, because of the reason that maximizing the margin is the object of the SVM. Thus we can generalize SVM RFE to the nonlinear case and other kernel methods. The following iterative procedure contains two steps proposed by Guyon et al. (2002):

The first step is to train SVM classifier, optimize

$$L_d(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) = \alpha^T \mathbf{1} - \frac{1}{2} \alpha^T H \alpha$$

*subject to*  $0 \leq \alpha_k \leq C, \sum_k \alpha_k y_k = 0$

(18)

where  $H = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ ,  $\mathbf{1} = [1, \dots, 1]^T$  ( $n$  by 1 vector).

The minimized cost function is (Vapnik, 1998)

$$J = \frac{1}{2} \alpha^{*T} H \alpha^* - \alpha^* \mathbf{1}$$
(19)

The second step is to compute  $DJ(i)$  for all features

$$DJ(i) = \frac{1}{2} \alpha^{*T} H \alpha^* - \frac{1}{2} \alpha^{*T} H(-i) \alpha^*$$
(20)

and then remove the feature with the smallest  $DJ(i)$ .

In the linear case,  $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$ ,  $\alpha^T H \alpha = \|\mathbf{w}\|^2$ . Therefore  $DJ(i) = \frac{1}{2} (w_i)^2$ , this is identical to linear SVMs in section 4.1.

(ii) First, we map the original data to a feature space of high or infinite dimensional space, and regard as a linear separable case in that space. Therefore we can use same criterion as in the sub-section 4.1 to select crucial genes. The drawback of this method is that we may generate

too many irrelevant genes. Taking the expression vector  $\mathbf{x}=[x_1 \ x_2 \ x_3]$  for example: If our dataset is second-order polynomial separable (in input space, the decision rule  $d(\mathbf{x})$  is a second-order polynomial), then we use the following mapping function  $\Phi(\mathbf{x})$ :

$$\Phi : \mathbf{x} \rightarrow \mathbf{z}(\mathbf{x}) = [x_1 \ x_2 \ x_3 \ x_1^2 \ x_2^2 \ x_3^2 \ x_1x_2 \ x_1x_3 \ x_2x_3],$$

We train this new expression vector  $\mathbf{z}(\mathbf{x})$  in SVM RFE with linear kernel to select crucial features (genes).

## 5 Simulation Studies

### 5.1 Linear problem

#### 5.1.1 Overlapping Classes

For training data sets without overlapping are rare in practice. In the following simulation, we study the case of overlapping classes in feature space.

We simulate *i.i.d.*  $x_{i,j} \sim N(0,1), i = 1, \dots, n, j = 1, \dots, 1024$  for training data set, representing microarray data with  $n$  subjects and the expression levels of 1024 genes, and *i.i.d.*  $x_{i,j} \sim N(0,1), i = 1, \dots, 1000, j = 1, \dots, 1024$  for test data set.

We define the Cauchy c.d.f. as  $F(x) = \frac{1}{\pi} (\tan^{-1}(\frac{x}{\sigma}) + \frac{\pi}{2}), \sigma = 0.25$  which is an increasing function and satisfies  $F(-\infty) = 0, F(0) = 1/2, F(\infty) = 1$ . Note that the larger the  $\sigma$ , the more the overlapping classes. Let  $x_1 + x_2 = s$ ,  $P(y = 1) = F(s)$ , and  $P(y = -1) = 1 - F(s)$ .

In the simulation, we use SVM to train  $\{x_{i,j}\}$  with linear kernel, RFE to eliminate gene one by one, and let  $C=100$ . Tables 1-6 give the results of the study. Table 1 give the gene selection results for training set size  $n=30, 40, 50$ , and 100. The table is the rankings in the first SVM training (with 1024 genes) and the reverse deleting order of gene1 and gene2. For instance, a reverse deleting order of 2 means the gene is the last one deleted, 1 means the gene

stays to the end, and “x” means the gene is deleted before the last 20 iterations. For n=30, the correct two features were selected about 1/5 times; for n=100, the correct two features were selected about 4/5 times. The results show that if n is larger, it is easier to select the crucial genes (gene1 and gene2) by SVMs with linear kernel for overlapping classes. Table 2 give the numbers of patients in class {+} and class {-} for various training set size. Tables 3-6 provide the training/test classification accuracy rate for the study.

### 5.1.2 Correlated Data

In this simulation study, we study the number of one gene’s surrogates somehow affect the importance of the gene.

We generate  $x_{i,j}, i = 1, \dots, 40, j = 1, \dots, p$   
 ( $p = 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000$ )

$x_{i,1}$ , with  $x_{i,2} \stackrel{i.i.d.}{\sim} N(0, 1), x_{i,j} \sim x_{i,2} + \varepsilon_{i,j}$  for  $j = 3, \dots, m \quad m = 3, \dots, 20,$

where  $\varepsilon_{i,j} \stackrel{i.i.d.}{\sim} N(0, 10^{-5}),$  and  $x_{i,m+1}, \dots, x_{i,p} \stackrel{i.i.d.}{\sim} N(0, 1),$  representing microarray data of 40 patients and  $p$  genes. The decision rule used in the study is: if  $x_1 + x_2 > 0$  then  $y \in class \{ +1 \},$  otherwise  $y \in class \{ -1 \}.$

For the simulation, we use SVM method to train  $\{x_{i,j}\}$  with linear kernel, RFE to eliminate gene one by one with C=100. Figure 1 gives the results of the study. The number of  $x_2$ 's surrogates is represented on the x-axis. The average of ( $x_2$ 's weight /  $x_1$ 's weight) and standard deviation over 50 random trials are represented on y-axis. We observe that these correlated genes with gene 2 may dilute the performance of gene 2. For the smaller  $p,$  the plot presents smooth decreasing concave curve; for the larger  $p,$  this is nearly a linear decreasing curve.

## 5.2 Nonlinear problem

### 5.2.1 Compare several different cases using nonlinear SVMs

In the following simulations, we study the effect of normalization for nonlinear classification problem and compare the performance of two different kernel functions: polynomial kernel of degree 2 and RBF kernel with parameter  $\sigma = 3$ . We simulate the independent training data  $x_{i,j}$ ,  $i = 1, \dots, 40$   $j = 1, \dots, 100$ , and test data  $x_{i,j}$ ,  $i = 1, \dots, 1000$   $j = 1, \dots, 100$ , representing microarray data with (40/1000) subjects and the expression levels of their 100 genes.

Our data set is generated from the following distributions:

- (i)  $N(0, 1)$
- (ii) Uniform  $(-0.5, 0.5)$
- (iii)  $N(0, 10)$
- (iv) Uniform  $(-10, 10)$
- (v)  $N(2, 1)$
- (vi) Uniform  $(1, 3)$



The following decision rules are used in the study:

- (a) decision rule: if  $x_1 x_2 > c$  then  $y \in class\{+\}$ , otherwise  $y \in class\{-\}$ ;
- (b) decision rule: if  $x_1^2 + x_2 > c$  then  $y \in class\{+\}$ , otherwise  $y \in class\{-\}$ ;
- (c) decision rule: if  $x_1^2 + x_2^2 > c$  then  $y \in class\{+\}$ , otherwise  $y \in class\{-\}$ ;
- (d) decision rule:  
if  $x_1^2 + x_1 x_2 + x_2^2 > c$  then  $y \in class\{+\}$ , otherwise  $y \in class\{-\}$ ;

where  $c$  is some constant, we choose it to balance the proportion of two classes.

For the study, we use the criterion in section 4.2 (i): nonlinear SVMs, eliminate gene one

by one, and  $C=100$ . For normalization, we discuss three different cases: normalizing  $\{x_{i,j}\}$  for each  $j$  (gene) then normalizing it for each  $i$  (subject); normalizing  $\{x_{i,j}\}$  only for each  $j$  (gene); not normalizing data. Tables 7-22 give the gene selection results of each case.

From Table 7, we observe the following:

- For the case of decision rule (a) and simulation data structure (i):  
With two degree polynomial and RBF kernels, we could not select crucial genes by our three normalization cases.
- For the case of decision rule (a) and simulation data structure (ii):  
With using polynomial kernel of degree 2, it seems normalizing data for each  $j$  (gene) then normalizing it for each  $i$  (subject), and normalizing data only for each  $j$  (gene) perform better than not normalizing data. With RBF kernel, it seems normalizing data only for each  $j$  (gene) performs better.
- For the case of decision rule (a) and simulation data structure (iii):  
With two degree polynomial and RBF kernels, it seems normalizing data for each  $j$  (gene) then normalizing it for each  $i$  (subject) performs better than other two cases.
- For the case of decision rule (a) and simulation data structure (iv):  
With two degree polynomial kernel, we all select crucial genes by three different normalization cases, but the gene selection results are not good with RBF kernel.
- For the case of decision rule (a) and simulation data structure (v) and (vi):  
With two degree polynomial kernel, it seems not normalizing data performs better than other two cases. With RBF kernel, three normalization cases are all satisfactory.

From Table 11, we observe the following:

- For the case of decision rule (b) and simulation data structure (i):  
With polynomial kernel of degree two, we only select gene1 by normalizing data for each  $j$  (gene) then normalizing it for each  $i$  (subject). With RBF kernel, we only select

gene2 by our three normalization cases.

- For the case of decision rule (b) and simulation data structure (ii):

With polynomial kernel of degree two, we only select gene2 by the case of not normalizing data. With RBF kernel, we only select gene2 by our three normalization cases.

- For the case of decision rule (b) and simulation data structure (iii):

With two degree polynomial and RBF kernels, we could not select crucial genes by our three normalization cases.

- For the case of decision rule (b) and simulation data structure (iv):

With polynomial kernel of degree two, we only select gene1 by our three normalization cases. With RBF kernel we could not select crucial genes by our three normalization cases.

- For the case of decision rule (b) and simulation data structure (v):

With two degree polynomial kernel, we could not select crucial genes by our three normalization cases. With RBF kernel we only select gene1 by our three normalization cases.

- For the case of decision rule (b) and simulation data structure (vi):

With two degree polynomial kernel, we could not select crucial genes by our three normalization cases. With RBF kernel, three normalization cases are all satisfactory.

From Table 15, we observe the following:

- For the case of decision rule (c) and simulation data structure (i) and (ii):

With two degree polynomial and RBF kernels, we could not select crucial genes by our three normalization cases.

- For the case of decision rule (c) and simulation data structure (iii):

With two degree polynomial kernel, we only select gene1 by normalizing data for each  $j$  (gene) then normalizing it for each  $i$  (subject). With RBF kernel, we could not select

crucial genes by our three normalization cases.

- For the case of decision rule (c) and simulation data structure (iv):

With two degree polynomial kernel, we could not select crucial genes by our three normalization cases. With RBF kernel, we only select gene1 by normalizing data for each  $j$  (gene).

- For the case of decision rule (c) and simulation data structure (v) and (vi):

With two degree polynomial kernel, it seems not normalizing data performs better than other two normalization cases. For RBF kernel, three normalization cases are all satisfactory.

From Table 19, we observe the following:

- For the case of decision rule (d) and simulation data structure (i), (ii), (iii) and (iv):

With two degree polynomial and RBF kernels, we could not select crucial genes by our three normalization cases.

- For the case of decision rule (d) and simulation data structure (v) and (vi):

With two degree polynomial kernel, it seems not normalizing data performs better than other two normalization cases. For RBF kernel, three normalization cases are all satisfactory.

Therefore, the choices of appropriate kernel functions are different for each type of decision rule and data structure. Tables 8, 12, 16, 20 give the number of patients in class  $\{+\}$  and class  $\{-\}$  of the training and testing datasets. Tables 9, 10, 13, 14, 17, 18, 21, 22 provide the training/test classification accuracy rate for the study.



## 5.2.2 Toy experiment

In this simulation study, we performed experiments on an example of nonlinear classification problem, the nonlinear toy problem provided in Weston *et al.* (2000).

Two features out of 52 are relevant. We utilize two methods described in sec.4.2 for the problem. The first method is the nonlinear kernel version of SVM RFE, we use a polynomial kernel of degree 2; The second method contains the following steps: we first map the data to higher dimensional space by a polynomial kernel of degree two, and then use SVM RFE with a linear kernel, see sec.4.2 (ii). With these two methods, we all normalize the data for each gene, then normalize it for each subject, and let  $C=100$ . The number of times of the correct features were selected over 30 random trials for various training set sizes with first method is shown in Table 23 and the times of the correct features were selected over 30 random trials for various training set sizes with second method is shown in Table 24. The classification performance (average test error on 500 examples over 30 random trials) of using these two methods is shown in Table 25 and Figure 2. From these results, we observe that the average performance of the second method is better than the first method for smaller training set size, but almost equally for larger training set size. In the first method, for  $n=10$  training examples, we selected average 25.3 features to obtain two relevant features; for  $n=100$ , an average of 2.17 features are selected to obtain two relevant features. In the second method, for  $n=10$ , we selected average 10.23 features to obtain the relevant feature  $(x_1 \times x_2)$ ; for  $n=100$ , an average of 1.33 features are selected to obtain relevant feature  $(x_1 \times x_2)$ . The results also show that these two methods are better than other feature selection methods using in Weston *et al.* (2000); Grandvalet, and Canu, (2002); Rakotomamonjy et al. (2003) for dealing the nonlinear toy problem.

## 6 Conclusion and Future Research

In our study, we review some literature about Support Vector Machine, which has shown great performance in practice as a classification methodology. In Sec.5, we experiment on linear and nonlinear classification problems utilizing SVMs.

From the simulation results of linear classification problem we observe that

- For overlapping classes, when the training set size is large, it is easy to select crucial genes by utilizing linear SVM RFE, and the performance of classification is also good.
- Numbers of one gene's surrogates would affect the importance of the gene.

From the simulation results of nonlinear classification problem we observe that

- The performance of classification is better with utilizing nonlinear SVM RFE criteria and our method (sec.4.2(ii)) than other feature selection methods utilized in Weston *et al.* (2000); Grandvalet, and Canu, (2002); Rakotomamonjy et al. (2003) for dealing the nonlinear toy problem.
- The drawback of the gene selection method described in sec.4.2 (ii) is that it is inappropriate to be utilized for large number of genes, but better (easy to select crucial genes) for small genes. Therefore we could combine other supervising learning methods with our method.

We propose a simple gene selection procedure for the case of nonlinear separable data. The procedure is applied to nonlinear toy problem. Many interesting problems are worth future study, such as finding a technique for choosing the kernel functions; finding other better feature selection methods; the choices of appropriate hyperparameters including degree  $d$  of a polynomial kernel, Gaussian kernel parameter  $\sigma$ , and penalized parameter  $C$ ; extending the binary SVM to the multcategory case. These problems are potential topics for future research.

## Reference

- [1] Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M., & Yakhini, Z. (2000). Tissue classification with gene expression profiles. *J. Computational Biology* 7, 559-584.
- [2] Boser, I. Guyon, and Vapnik, V. (1992). An training algorithm for optimal classifiers. *Fifth Annual Workshop on Computational Learning Theory, Pittsburgh ACM*, pp. 144-152.
- [3] Cristianini, N., Campbel, C., and Shawe-Taylor, J. (1998). Dynamically adapting kernels in support vector machines. In *Advances in Neural Information Processing Systems*.
- [4] Fujarewicz, K., Wiench, M. (2003). Selecting differentially expressed genes for colon tumor classification, *Int. J. Appl. Math. Comput. Sci.*, Vol. 13, No. 3, 327-335.
- [5] Furey, T., Cristianini, N., Duffy, N., Bednarski, D., Schummer, M., & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16, 906-914.
- [6] Golub, T., Slonim, D., tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloom- field, C., and Lander, E. (1999). Molecular classification of cancer; class discovery and prediction by gene expression monitoring. *Science* **286**, 531-537. The data is available on-line at [http://www-genome.wi.mit.edu/MPR/data\\_set\\_ALL\\_AML.html](http://www-genome.wi.mit.edu/MPR/data_set_ALL_AML.html).
- [7] Grandvalet, Y. and Canu, S. (2002). Adaptive scaling for feature selection in SVMs. In *NIPS* 15.
- [8] GUNN, S. R. (1998). Support Vector Machines for Classification and Regression. *Technical Report*, Image Speech and Intelligent Systems Research Group, University of Southampton.

- [9] Guyon, I., Weston, J., Barnhill, S., and Vapnik V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning* **46**, 389-422.
- [10] Khan, J., Wei, J., Ringner, M., Atonescu, C., Peterson, C. and Meltzer, P. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine* **7**, 673-679.
- [11] Krishnapuram, B., Carin, L., Hartemink, A. (2004). Gene Expression Analysis: Joint Feature Selection and Classifier Design. In *Kernel Methods in Computational Biology*, Schölkopf, B., Tsuda, K., & Vert, J.-P., eds. MIT Press.
- [12] Lee, Y., Lin, Y., and Wahba, G. (2001). Multicategory Support Vector Machines. *Proceedings of the 33<sup>rd</sup> Symposium on the Interface*. Also available as *TR 1043, Statistics Dept., University of Wisconsin-Madison*.
- [13] Lee, Y., and Lee, C. (2002). Classification of multiple cancer types by multicategory support vector machines using gene expression data. *TR 1051; Statistical Dept., University of Wisconsin-Madison. To appear in Bioinformatics*.
- [14] Markowetz, F., Edler, L., and Vingron, M. (2003). Support Vector Machines for Protein Fold Class Prediction. *Biometrical Journal* **45**, 3, 377-389
- [15] Pavlidis, P., Weston, J., Cai, J., & Grundy, W. N. (2000). Gene functional analysis from heterogeneous data. Submitted for publication.
- [16] Rakotomamonjy, A. (2003). Variable Selection Using SVM-based Criteria. *Journal of Machine Learning research*, 3:1357-1370.
- [17] Rifkin, R. M. (2002). *Everything Old Is New Again: A Fresh Look at Historical Approaches in Machine Learning*. Massachusetts Institute of Technology.
- [18] Vapnik, V. (1998). *Statistical Learning Theory*. New York, Wiley.

- [19] Weston, J., Muckerjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. (2000). Feature selection for SVMs. *Advances in Neural Information Processing Systems*.
- [20] Zhang, X. and Wong, W. H. (2001). Recursive Sample Classification and Gene Selection based on SVM: Method and Software Description, Biostatistics Dpt. Tech Report, *Harvard School of Public Health*.



Table 1: The rankings of gene1 and gene2 in the first SVM training (with 1024 genes) and the reverse deleting order by RFE and the two genes stay to the end (overlapping classes), n is the training set size.

n	Step	gene1	gene2	gene1	gene2	gene1	gene2	gene1	gene2	gene1	gene2
30	first ranking	46	1	2	1	12	4	4	1	6	1
	rev-del order	x	4	6	1	x	14	1	2	15	11
	last 2 genes	993	290	2	259	518	906	1	2	746	736
40	first ranking	1	107	5	6	5	20	1	7	4	9
	rev-del order	1	x	2	1	1	2	1	2	x	18
	last 2 genes	1	585	2	1	1	2	1	2	495	195
50	first ranking	4	1	89	1	5	1	1	2	1	2
	rev-del order	2	1	x	1	9	10	1	2	9	8
	last 2 genes	2	1	2	453	776	894	1	2	796	685
100	first ranking	2	1	1	2	1	2	1	3	1	2
	rev-del order	2	1	1	2	2	1	1	5	2	1
	last 2 genes	2	1	1	2	2	1	1	721	2	1

Table 2: The number of patients in positive (+) class and negative (-) class of the training and testing datasets.

n	data set	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)
30	train	15	15	20	10	14	16	17	13	18	12
	test	503	497	523	477	481	519	501	499	487	513
40	train	20	20	22	18	19	21	26	14	22	18
	test	502	498	506	494	512	488	494	506	491	509
50	train	26	24	26	24	26	24	25	25	27	23
	test	520	480	493	507	518	482	508	492	500	500
100	train	51	49	55	45	49	51	47	53	55	45
	test	506	494	508	492	493	507	511	489	481	519

Table 3: The (training/testing) classification accuracy rate, the first column is the number of genes still in the training set. The training set size is 30 examples, and the test set size is 1000 examples.

# genes still in the training set	accuracy rate (n=30)				
1024	1/0.505	1/0.521	1/0.546	1/0.540	1/0.522
512	1/0.506	1/0.521	1/0.543	1/0.541	1/0.527
256	1/0.509	1/0.530	1/0.540	1/0.537	1/0.530
128	1/0.535	1/0.536	1/0.547	1/0.570	1/0.537
64	1/0.574	1/0.557	1/0.553	1/0.570	1/0.564
32	1/0.582	1/0.579	1/0.556	1/0.580	1/0.600
16	1/0.547	1/0.626	1/0.518	1/0.588	1/0.593
8	1/0.597	1/0.615	1/0.462	1/0.618	1/0.505
4	1/0.592	1/0.623	1/0.492	1/0.792	1/0.502
3	0.9/0.537	0.933/0.651	1/0.492	0.967/0.815	1/0.509
2	0.833/0.517	0.9/0.677	0.867/0.482	0.9/0.855	0.867/0.517
1	0.7/0.537	0.867/0.733	0.767/0.468	0.767/0.718	0.833/0.499

Table 4: The (training/testing) classification accuracy rate, the first column is the number of genes still in the training set. The training set size is 40 examples, and the test set size is 1000 examples.

# genes still in the training set	accuracy rate (n=40)				
1024	1/0.506	1/0.514	1/0.527	1/0.505	1/0.526
512	1/0.510	1/0.519	1/0.526	1/0.507	1/0.525
256	1/0.543	1/0.508	1/0.526	1/0.510	1/0.535
128	1/0.544	1/0.526	1/0.552	1/0.529	1/0.531
64	1/0.565	1/0.534	1/0.556	1/0.546	1/0.542
32	1/0.564	1/0.559	1/0.588	1/0.585	1/0.557
16	1/0.559	1/0.645	1/0.662	1/0.613	1/0.491
8	1/0.566	1/0.660	1/0.658	1/0.641	1/0.510
4	0.975/0.618	0.975/0.768	1/0.735	0.95/0.698	1/0.494
3	0.875/0.635	0.85/0.797	0.9/0.821	0.875/0.781	0.875/0.492
2	0.8/0.683	0.85/0.877	0.9/0.864	0.875/0.839	0.8/0.471
1	0.8/0.719	0.625/0.739	0.725/0.704	0.75/0.678	0.675/0.466

Table 5: The (training/testing) classification accuracy rate, the first column is the number of genes still in the training set. The training set size is 50 examples, and the test set size is 1000 examples.

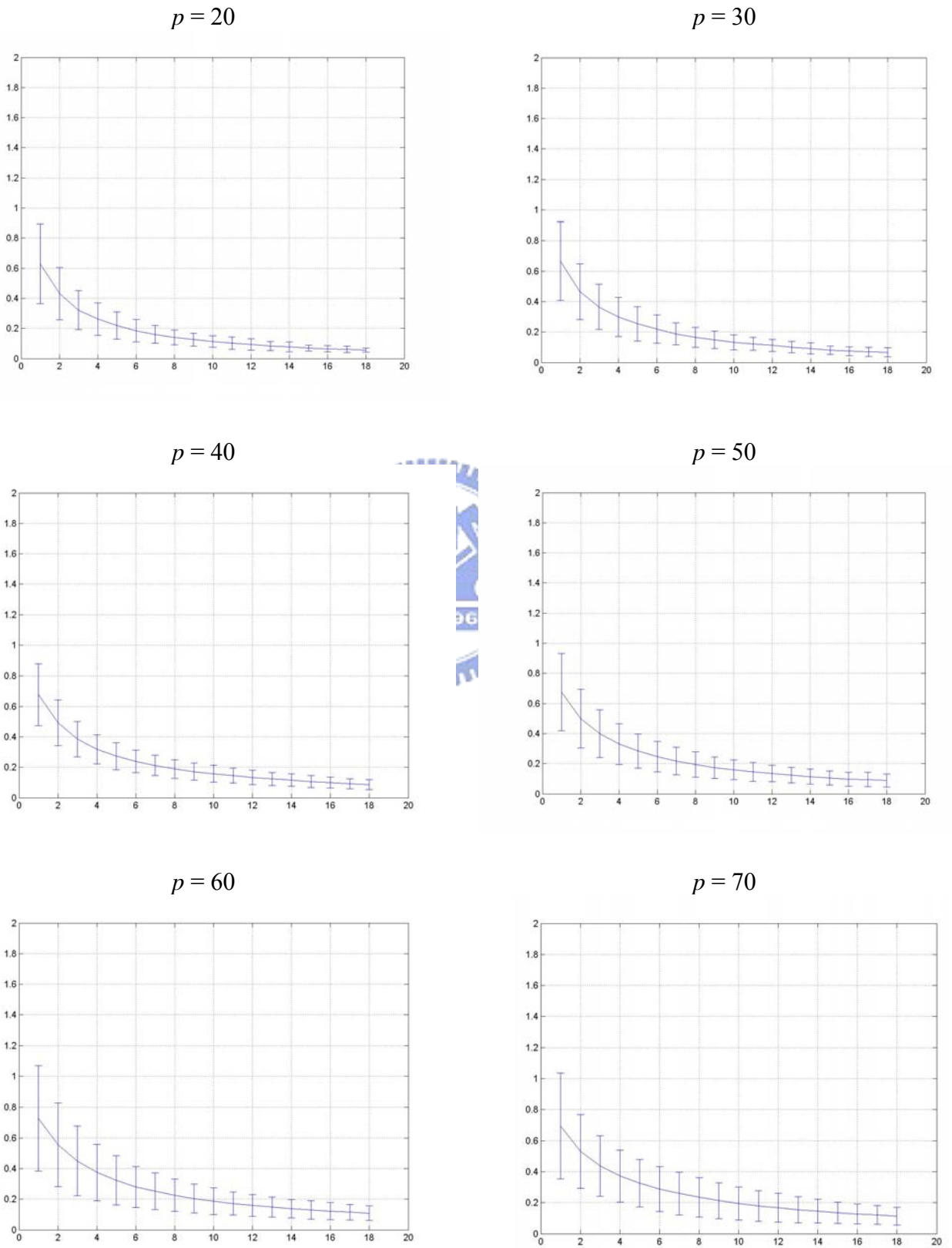
# genes still in the training set	accuracy rate (n=50)				
1024	1/0.566	1/0.557	1/0.542	1/0.542	1/0.535
512	1/0.568	1/0.558	1/0.557	1/0.541	1/0.541
256	1/0.575	1/0.562	1/0.570	1/0.568	1/0.536
128	1/0.578	1/0.559	1/0.563	1/0.586	1/0.545
64	1/0.603	1/0.536	1/0.567	1/0.598	1/0.567
32	1/0.616	1/0.554	1/0.569	1/0.593	1/0.597
16	1/0.624	1/0.555	1/0.575	1/0.601	1/0.605
8	1/0.701	1/0.579	1/0.509	1/0.756	1/0.551
4	0.98/0.816	0.94/0.613	0.86/0.502	1/0.791	0.94/0.493
3	0.96/0.798	0.92/0.618	0.84/0.499	0.96/0.831	0.8/0.492
2	0.9/0.861	0.8/0.651	0.8/0.473	0.92/0.869	0.82/0.481
1	0.8/0.689	0.74/0.72	0.8/0.49	0.84/0.72	0.68/0.497

Table 6: The (training/testing) classification accuracy rate, the first column is the number of genes still in the training set. The training set size is 100 examples, and the test set size is 1000 examples.

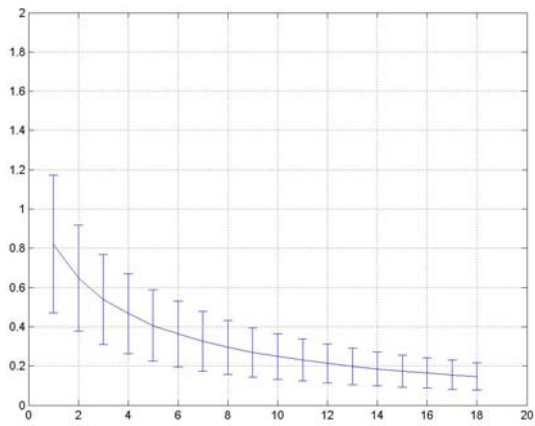
# genes still in the training set	accuracy rate (n=100)				
1024	1/0.578	1/0.561	1/0.572	1/0.536	1/0.559
512	1/0.553	1/0.553	1/0.574	1/0.551	1/0.559
256	1/0.570	1/0.557	1/0.591	1/0.548	1/0.551
128	1/0.561	1/0.6	1/0.583	1/0.558	1/0.555
64	1/0.583	1/0.621	1/0.633	1/0.577	1/0.564
32	1/0.614	1/0.643	1/0.670	1/0.590	1/0.602
16	1/0.690	1/0.669	1/0.659	1/0.631	1/0.662
8	0.97/0.723	1/0.723	0.94/0.709	0.94/0.687	0.93/0.675
4	0.88/0.831	0.92/0.86	0.92/0.804	0.87/0.655	0.92/0.775
3	0.91/0.831	0.93/0.858	0.87/0.851	0.8/0.644	0.87/0.779
2	0.86/0.872	0.92/0.882	0.92/0.873	0.8/0.665	0.84/0.833
1	0.72/0.691	0.72/0.711	0.74/0.717	0.75/0.712	0.65/0.715



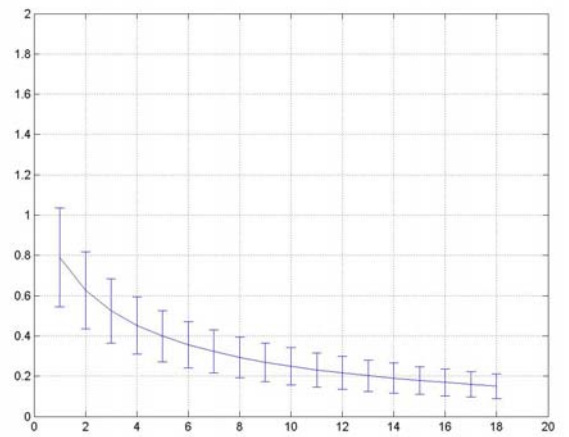
Figure 1: The number of  $x_2$ 's surrogates is represented on the x-axis. The average of ( $x_2$ 's weight /  $x_1$ 's weight) and standard deviation over 50 random trials are represented on y-axis.



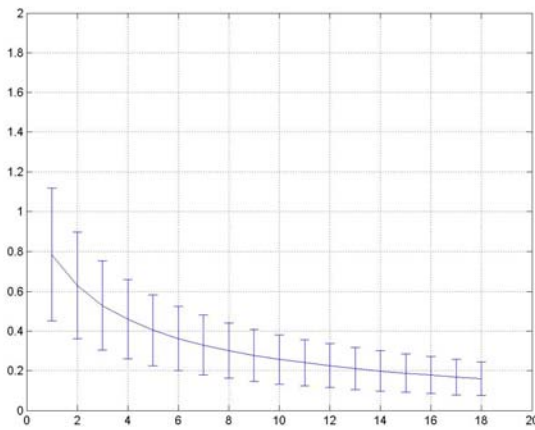
$p = 80$



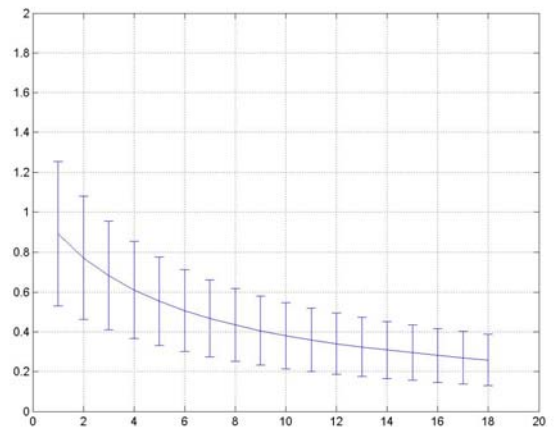
$p = 90$



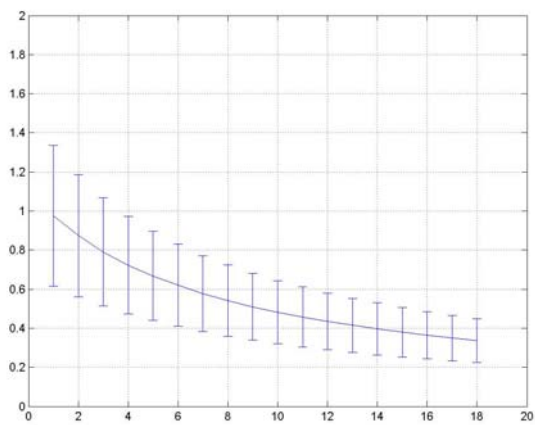
$p = 100$



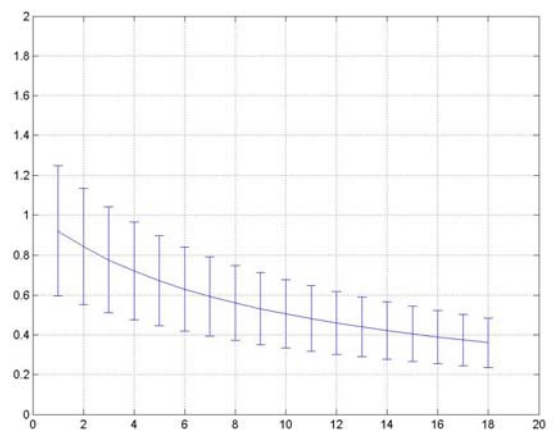
$p = 200$



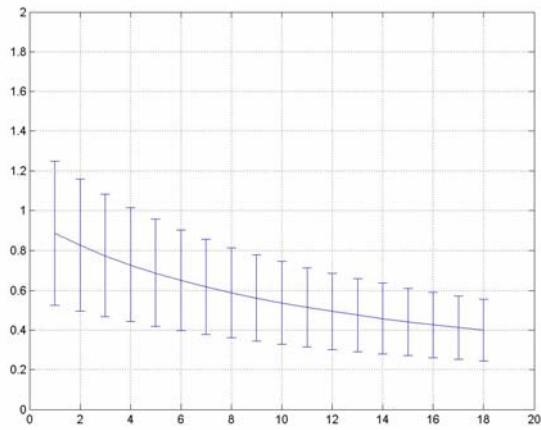
$p = 300$



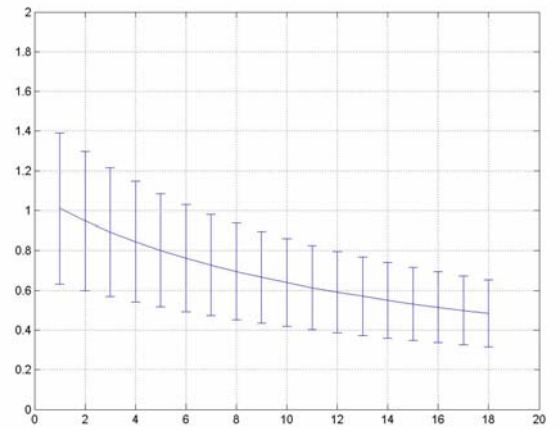
$p = 400$



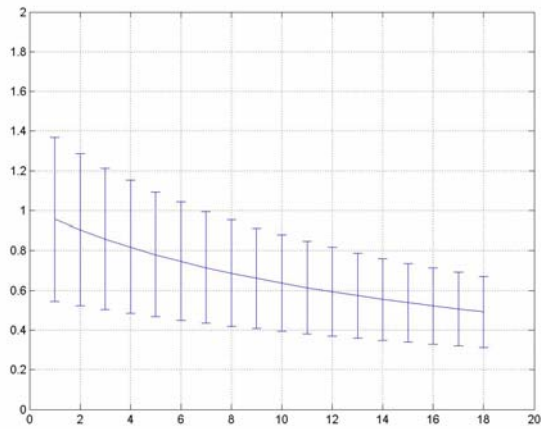
$p = 500$



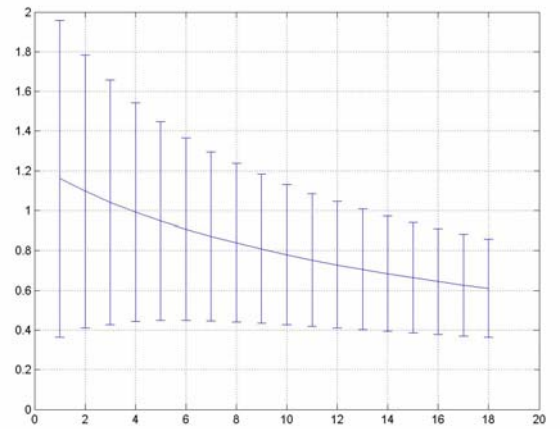
$p = 600$



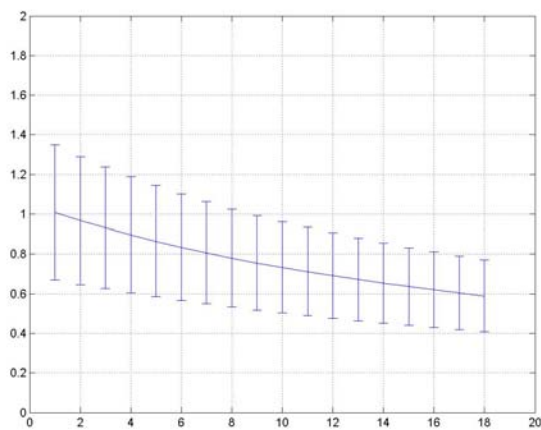
$p = 700$



$p = 800$



$p = 900$



$p = 1000$

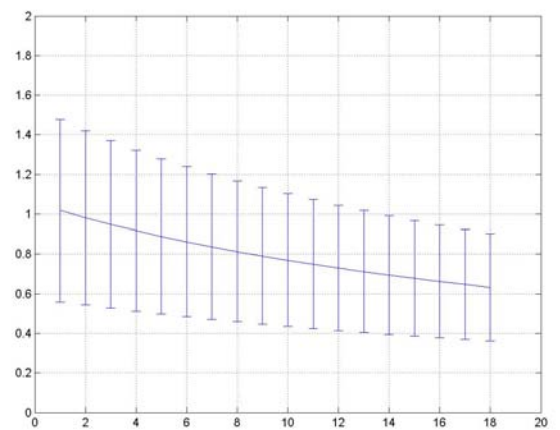


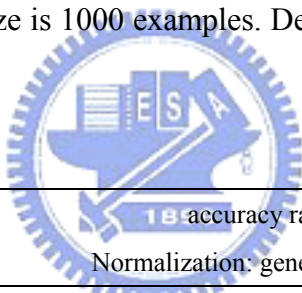
Table 7: The rankings of gene1 and gene2 in the first training (with 100 genes) and the reverse deleting by nonlinear SVM RFE with a polynomial kernel of degree 2 and RBF kernel with parameter  $\sigma = 3$  and the two genes stay to the end (Decision function  $x_1x_2 = c$ , six kinds of different data structures, and three different normalization cases ).

c	0	0	0	0	2.5	4
data	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
poly-2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2
Normalizing (gene →subject)						
first ranking	67 65	5 18	16 51	47 8	99 30	100 92
rev-del order	63 64	1 2	2 1	1 2	99 44	100 92
last two genes	52 49	<b>1 2</b>	<b>2 1</b>	<b>1 2</b>	50 37	85 45
Normalize (gene)						
first ranking	69 73	1 16	25 83	33 3	95 33	100 97
rev-del order	80 81	1 2	64 76	2 1	96 3	100 94
last two genes	79 77	<b>1 2</b>	22 23	<b>2 1</b>	29 69	60 72
not normalizing data						
first ranking	81 90	3 94	70 41	7 19	1 10	2 1
rev-del order	89 91	17 94	73 58	2 1	5 1	1 3
last two genes	100 35	65 54	37 7	<b>2 1</b>	<b>2 3</b>	<b>1 20</b>
rbf-3	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2
Normalizing (gene →subject)						
first ranking	29 31	3 22	16 39	51 12	1 3	2 1
rev-del order	20 19	1 35	2 1	24 23	3 2	1 2
last two genes	92 57	<b>1 21</b>	<b>2 1</b>	23 16	9 <b>2</b>	<b>1 2</b>
Normalizing (gene)						
first ranking	29 49	2 37	30 19	30 7	1 14	2 1
rev-del order	10 9	2 1	26 25	24 23	1 2	1 2
last two genes	6 61	<b>2 1</b>	22 23	16 3	<b>1 2</b>	<b>1 2</b>
not normalizing data						
first ranking	32 38	10 66	100 99	100 99	1 18	2 1
rev-del order	23 15	10 52	100 99	100 99	1 2	2 1
last two genes	87 84	4 63	85 97	92 81	<b>1 2</b>	<b>2 1</b>

Table 8: The number of subjects in positive (+) class and negative (-) class of the training and testing datasets.

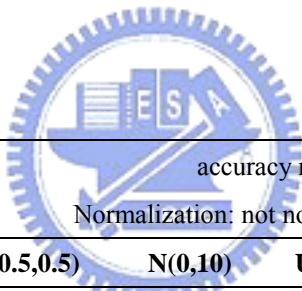
Data set												
	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)
train	20	20	21	19	24	16	19	21	20	20	19	21
test	491	509	498	502	492	508	483	517	646	354	448	552

Table 9: The (training/testing) classification accuracy rate for three different normalization cases, the first column is the number of genes still in the training set. The training set size is 40 examples, and the test set size is 1000 examples. Decision function:  $x_1x_2 = c$ . Kernel: two degree polynomial.



# gene selected	accuracy rate					
	Normalization: gene $\rightarrow$ subject					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.522	1/0.501	1/0.508	1/0.499	1/0.485	1/0.522
50	1/0.494	1/0.528	1/0.497	1/0.521	1/0.501	1/0.507
25	1/0.506	1/0.583	1/0.524	1/0.573	1/0.460	1/0.482
12	1/0.494	1/0.606	1/0.580	1/0.587	1/0.486	1/0.496
6	1/0.515	1/0.713	1/0.657	1/0.685	1/0.493	1/0.471
5	1/0.512	1/0.708	1/0.633	1/0.689	1/0.460	1/0.489
4	0.975/0.534	1/0.783	1/0.634	1/0.759	1/0.472	1/0.487
3	0.8/0.515	1/0.855	1/0.833	1/0.801	0.85/0.448	0.825/0.484
2	0.775/0.516	0.975/0.876	0.975/0.775	0.95/0.923	0.7/0.424	0.7/0.485
1	0.7/0.511	0.65/0.481	0.625/0.493	0.625/0.502	0.55/0.566	0.575/0.512

# gene selected	accuracy rate					
	Normalization: gene					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.512	1/0.508	1/0.504	1/0.494	1/0.476	1/0.519
50	1/0.494	1/0.539	1/0.502	1/0.531	1/0.493	1/0.505
25	1/0.499	1/0.559	1/0.494	1/0.567	1/0.498	1/0.475
12	1/0.488	1/0.609	1/0.500	1/0.594	1/0.546	1/0.506
6	1/0.489	1/0.709	1/0.484	1/0.721	1/0.567	1/0.507
5	1/0.470	1/0.745	1/0.499	1/0.768	1/0.546	1/0.502
4	0.825/0.502	1/0.836	1/0.480	1/0.809	1/0.512	0.95/0.512
3	0.8/0.493	1/0.832	0.8/0.508	1/0.869	0.825/0.524	0.825/0.499
2	0.675/0.514	1/0.865	0.8/0.504	1/0.930	0.7/0.491	0.775/0.483
1	0.6/0.483	0.625/0.484	0.675/0.487	0.6/0.497	0.675/0.474	0.65/0.524



# gene selected	accuracy rate					
	Normalization: not normalizing data					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.526	1/0.503	1/0.495	1/0.500	1/0.593	1/0.655
50	1/0.486	1/0.483	1/0.481	1/0.543	1/0.602	1/0.672
25	1/0.500	1/0.501	1/0.486	1/0.559	1/0.692	1/0.690
12	1/0.486	1/0.484	1/0.459	1/0.649	1/0.730	1/0.751
6	1/0.500	1/0.495	1/0.490	1/0.733	1/0.853	1/0.740
5	1/0.484	0.975/0.476	1/0.480	1/0.799	1/0.853	1/0.779
4	0.95/0.508	0.9/0.486	0.925/0.499	1/0.908	0.95/0.557	1/0.929
3	0.85/0.509	0.775/0.490	0.7/0.497	1/0.954	0.85/0.548	1/0.941
2	0.725/0.484	0.725/0.498	0.7/0.494	1/0.966	0.775/0.652	0.8/0.735
1	0.575/0.485	0.65/0.47	0.6/0.492	0.6/0.495	0.725/0.745	0.8/0.743

Table 10: The (training/testing) classification accuracy rate for three different normalization cases, the first column is the number of genes still in the training set. The training set size is 40 examples, and the test set size is 1000 examples. Decision function:  $x_1x_2 = c$ . Kernel: RBF

# gene selected	accuracy rate					
	Normalization: gene $\rightarrow$ subject					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.516	1/0.506	1/0.504	1/0.510	1/0.592	1/0.631
50	1/0.504	1/0.521	1/0.493	1/0.516	1/0.654	1/0.654
25	1/0.508	1/0.505	1/0.499	1/0.522	1/0.68	1/0.702
12	1/0.501	1/0.497	1/0.502	1/0.491	1/0.673	1/0.765
6	1/0.500	1/0.509	1/0.630	1/0.474	1/0.689	1/0.825
5	1/0.486	1/0.510	0.975/0.672	1/0.510	1/0.750	1/0.854
4	0.9/0.494	0.875/0.508	0.975/0.674	0.975/0.499	1/0.854	1/0.899
3	0.75/0.493	0.85/0.496	0.925/0.733	0.875/0.52	0.95/0.865	1/0.928
2	0.675/0.488	0.725/0.499	0.9/0.785	0.675/0.528	0.8/0.595	1/0.929
1	0.575/0.477	0.65/0.482	0.625/0.503	0.65/0.522	0.65/0.5	0.8/0.743

# gene selected	accuracy rate					
	Normalization: gene					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.509	1/0.499	1/0.492	1/0.503	1/0.549	1/0.628
50	1/0.518	1/0.507	1/0.488	1/0.529	1/0.613	1/0.66
25	1/0.491	1/0.501	1/0.489	1/0.53	1/0.629	1/0.693
12	1/0.513	1/0.526	1/0.484	1/0.493	1/0.671	1/0.729
6	1/0.513	1/0.663	1/0.489	1/0.496	1/0.724	1/0.777
5	1/0.505	0.975/0.667	0.975/0.482	1/0.504	1/0.784	1/0.907
4	0.95/0.497	0.95/0.845	0.925/0.505	0.95/0.496	1/0.880	1/0.905
3	0.825/0.509	0.95/0.841	0.9/0.501	0.875/0.499	0.975/0.877	1/0.971
2	0.75/0.499	0.95/0.837	0.8/0.501	0.725/0.495	1/0.912	1/0.97
1	0.625/0.488	0.6/0.498	0.65/0.498	0.6/0.515	0.775/0.708	0.8/0.743

# gene selected	accuracy rate					
	Normalization: not normalizing data					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.507	1/0.492	1/0.464	1/0.499	1/0.561	1/0.638
50	1/0.515	1/0.500	1/0.497	1/0.491	1/0.655	1/0.667
25	1/0.509	1/0.482	1/0.483	1/0.486	1/0.638	1/0.691
12	1/0.498	1/0.476	1/0.498	1/0.524	1/0.735	1/0.758
6	1/0.513	0.95/0.483	1/0.503	1/0.506	1/0.746	1/0.884
5	0.95/0.488	0.925/0.488	1/0.506	1/0.501	1/0.805	1/0.915
4	0.925/0.485	0.8/0.488	1/0.487	1/0.500	1/0.923	1/0.951
3	0.8/0.516	0.7/0.490	1/0.510	1/0.474	0.975/0.917	0.975/0.959
2	0.625/0.499	0.625/0.497	1/0.500	0.975/0.494	1/0.975	0.975/0.982
1	0.55/0.495	0.675/0.5	0.7/0.495	0.7/0.491	0.775/0.75	0.775/0.745





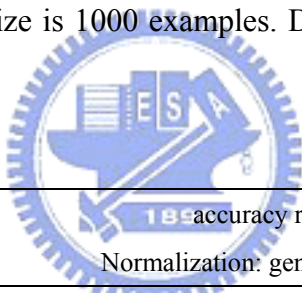
Table 11: The rankings of gene1 and gene2 in the first training (with 100 genes) and the reverse deleting by nonlinear SVM RFE with a polynomial kernel of degree 2 and RBF kernel with parameter  $\sigma = 3$  and the two genes stay to the end (Decision function  $x_1^2 + x_2 = c$ , six kinds of different data structures, and three different normalization cases ).

c	0.5	0	40	30	6	6
data	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
(poly-2)	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2
Normalize (gene → subject)						
first ranking	53 93	98 90	45 42	37 54	99 68	100 88
rev-del order	1 92	98 93	70 6	1 34	99 33	100 93
last two genes	<b>1</b> 33	72 77	60 18	<b>1</b> 20	27 91	42 50
Normalize (gene)						
first ranking	57 94	97 82	48 83	26 62	95 67	100 92
rev-del order	21 95	96 85	47 31	1 52	96 54	100 94
last two genes	78 36	93 41	6 60	<b>1</b> 4	15 33	63 62
not normalizing data						
first ranking	78 49	93 1	87 57	66 10	1 19	1 3
rev-del order	64 50	86 1	91 48	1 66	3 5	3 7
last two genes	100 77	<b>2</b> 65	161 94	<b>1</b> 54	100 41	53 82
(rbf-3)	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2
normalize (gene → subject)						
first ranking	95 1	99 1	81 6	72 82	1 69	1 5
rev-del order	98 1	99 1	72 11	62 71	1 46	1 4
last two genes	<b>2</b> 100	<b>2</b> 4	12 18	76 73	<b>1</b> 25	<b>1</b> 24
normalize(gene)						
first ranking	90 1	100 1	68 60	93 91	1 65	1 8
rev-del order	95 1	100 1	88 6	69 77	1 66	1 2
last two genes	<b>2</b> 3	<b>2</b> 4	18 60	64 67	<b>1</b> 25	<b>1</b> 2
not normalizing data						
first ranking	90 1	89 1	100 99	100 99	1 67	1 5
rev-del order	77 1	91 1	100 99	100 99	1 65	1 6
last two genes	<b>2</b> 71	<b>2</b> 4	96 90	76 100	<b>1</b> 34	<b>1</b> 96

Table 12: The number of subjects in positive (+) class and negative (-) class of the training and testing datasets.

Data set												
	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)
train	24	16	27	13	20	20	21	19	14	26	21	19
test	586	414	566	434	518	482	431	569	507	493	507	493

Table 13: The (training/testing) classification accuracy rate for three different normalization cases, the first column is the number of genes still in the training set. The training set size is 40 examples, and the test set size is 1000 examples. Decision function:  $x_1^2 + x_2 > c$ . Kernel: two degree polynomial.



# gene selected	accuracy rate					
	Normalization: gene → subject					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.560	1/0.556	1/0.475	1/0.494	1/0.542	1/0.513
50	1/0.533	1/0.532	1/0.475	1/0.505	1/0.524	1/0.496
25	1/0.531	1/0.526	1/0.481	1/0.524	1/0.52	1/0.472
12	1/0.550	1/0.528	1/0.490	1/0.553	1/0.51	1/0.487
6	1/0.568	1/0.504	1/0.493	1/0.651	1/0.527	1/0.492
5	1/0.565	1/0.510	1/0.490	1/0.686	1/0.510	1/0.504
4	0.975/0.589	1/0.509	1/0.487	1/0.783	0.975/0.505	1/0.513
3	0.825/0.637	0.95/0.534	0.975/0.484	1/0.881	0.875/0.498	0.85/0.523
2	0.75/0.683	0.775/0.528	0.75/0.507	1/0.91	0.85/0.506	0.725/0.508
1	0.775/0.69	0.675/0.566	0.525/0.527	0.975/0.912	0.675/0.484	0.675/0.471

# gene selected	accuracy rate					
	Normalization: gene					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.552	1/0.549	1/0.488	1/0.482	1/0.542	1/0.511
50	1/0.531	1/0.534	1/0.522	1/0.505	1/0.515	1/0.488
25	1/0.521	1/0.563	1/0.495	1/0.535	1/0.495	1/0.507
12	1/0.53	1/0.493	1/0.499	1/0.553	1/0.494	1/0.499
6	1/0.511	1/0.508	1/0.518	1/0.650	1/0.493	1/0.485
5	1/0.515	1/0.509	1/0.503	1/0.647	1/0.489	1/0.483
4	1/0.513	0.975/0.515	0.95/0.485	1/0.766	0.975/0.505	1/0.495
3	0.75/0.522	0.775/0.539	0.775/0.499	1/0.857	0.95/0.492	0.875/0.492
2	0.725/0.529	0.725/0.54	0.725/0.51	0.95/0.939	0.675/0.498	0.75/0.513
1	0.6/0.586	0.675/0.566	0.475/0.5	0.95/0.936	0.65/0.493	0.625/0.516

# gene selected	accuracy rate					
	Normalization: not normalizing data					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.545	1/0.602	1/0.495	1/0.482	1/0.628	1/0.669
50	1/0.527	1/0.636	1/0.493	1/0.496	1/0.641	1/0.666
25	1/0.481	1/0.697	1/0.504	1/0.547	1/0.68	1/0.705
12	1/0.472	1/0.792	1/0.478	1/0.557	1/0.746	1/0.764
6	1/0.485	1/0.812	1/0.502	1/0.693	1/0.777	1/0.843
5	1/0.497	1/0.868	1/0.471	1/0.781	1/0.764	1/0.876
4	0.975/0.489	1/0.893	0.925/0.5	1/0.851	1/0.888	1/0.883
3	0.85/0.523	1/0.895	0.775/0.482	1/0.786	1/0.897	1/0.890
2	0.825/0.526	0.975/0.937	0.75/0.487	1/0.924	0.675/0.503	0.7/0.5
1	0.725/0.536	0.925/0.939	0.625/0.504	0.975/0.958	0.65/0.493	0.675/0.513

Table 14: The (training/testing) classification accuracy rate for three different normalization cases, the first column is the number of genes still in the training set. The training set size is 40 examples, and the test set size is 1000 examples. Decision function:  $x_1^2 + x_2 > c$ . Kernel: RBF

# gene selected	accuracy rate					
	Normalization: gene $\rightarrow$ subject					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.599	1/0.590	1/0.508	1/0.521	1/0.552	1/0.67
50	1/0.588	1/0.576	1/0.478	1/0.469	1/0.499	1/0.686
25	1/0.655	1/0.663	1/0.503	1/0.488	1/0.603	1/0.750
12	1/0.679	1/0.746	1/0.502	1/0.496	1/0.680	1/0.758
6	1/0.681	1/0.764	1/0.472	1/0.507	1/0.675	1/0.767
5	1/0.689	1/0.779	1/0.473	1/0.500	1/0.759	1/0.834
4	1/0.685	1/0.785	0.95/0.46	0.975/0.515	1/0.791	1/0.865
3	0.975/0.761	1/0.791	0.775/0.458	0.925/0.508	1/0.807	0.975/0.881
2	0.925/0.769	1/0.810	0.75/0.442	0.75/0.499	0.975/0.792	0.925/0.925
1	0.85/0.781	0.925/0.915	0.625/0.48	0.625/0.487	0.975/0.802	0.9/0.936

# gene selected	accuracy rate					
	Normalization: gene					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.595	1/0.587	1/0.506	1/0.494	1/0.550	1/0.674
50	1/0.587	1/0.583	1/0.517	1/0.462	1/0.496	1/0.725
25	1/0.660	1/0.674	1/0.514	1/0.487	1/0.605	1/0.743
12	1/0.673	1/0.732	1/0.510	1/0.507	1/0.683	1/0.833
6	1/0.660	1/0.777	1/0.514	1/0.494	1/0.690	1/0.852
5	1/0.685	1/0.785	1/0.486	1/0.492	1/0.760	1/0.892
4	1/0.652	1/0.806	0.95/0.494	0.925/0.475	1/0.779	1/0.898
3	0.95/0.718	1/0.836	0.875/0.483	0.8/0.501	1/0.794	1/0.898
2	0.875/0.77	1/0.834	0.725/0.495	0.75/0.462	0.975/0.774	1/0.937
1	0.85/0.783	0.925/0.909	0.525/0.514	0.625/0.521	0.975/0.808	0.925/0.939

# gene selected	accuracy rate					
	Normalization: not normalizing data					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.592	1/0.624	1/0.580	1/0.500	1/0.569	1/0.692
50	1/0.588	1/0.657	1/0.480	1/0.480	1/0.508	1/0.717
25	1/0.657	1/0.675	1/0.502	1/0.476	1/0.632	1/0.765
12	1/0.700	1/0.728	1/0.498	1/0.497	1/0.701	1/0.887
6	1/0.747	1/0.815	1/0.487	1/0.480	1/0.832	1/0.866
5	1/0.716	1/0.887	1/0.506	1/0.469	1/0.840	1/0.865
4	1/0.717	1/0.886	1/0.500	1/0.480	1/0.833	0.925/0.896
3	0.975/0.741	0.975/0.881	1/0.513	1/0.504	1/0.828	0.925/0.931
2	0.925/0.747	1/0.89	1/0.501	0.95/0.506	1/0.893	10.925/0.932
1	0.9/0.774	0.95/0.936	0.85/0.505	0.75/0.491	0.975/0.91	0.925/0.945



Table 15: The rankings of gene1 and gene2 in the first training (with 100 genes) and the reverse deleting by nonlinear SVM RFE with a polynomial kernel of degree 2 and RBF kernel with parameter  $\sigma = 3$  and the two genes stay to the end (Decision function  $x_1^2 + x_2^2 = c$ , six kinds of different data structures, and three different normalization cases ).

c	1	0.1	100	60	8	8
data	N(0,1)	Uni(-0.5,0.5)	N(0,10)	Uni(-10,10)	N(2,1)	Uni(1,3)
poly-2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2
Normalize (gene → subject)						
first ranking	41 37	50 89	28 69	65 4	32 76	80 98
rev-del order	46 52	43 89	1 29	85 32	64 49	73 98
last two genes	65 45	94 61	<b>1</b> 55	27 57	65 25	14 5
Normalize (gene)						
first ranking	57 48	39 84	35 79	51 5	25 69	82 97
rev-del order	68 62	27 83	37 53	71 7	31 1	81 97
last two genes	20 88	65 61	22 55	50 64	<b>2</b> 86	14 5
not normalizing						
first ranking	81 93	43 91	79 33	30 28	3 1	2 1
rev-del order	77 94	38 91	81 29	33 41	5 4	1 3
last two genes	35 89	98 20	15 8	54 17	90 42	<b>1</b> 67
rbf-3	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2
Normalize (gene → subject)						
first ranking	100 92	87 89	79 99	62 8	5 1	4 1
rev-del order	95 100	77 64	93 99	75 76	2 1	2 1
last two genes	67 40	88 87	31 6	38 25	<b>2</b> <b>1</b>	<b>2</b> <b>1</b>
Normalize (gene)						
first ranking	59 57	88 57	63 99	39 29	5 1	3 1
rev-del order	28 30	77 54	68 70	1 87	2 1	4 1
last two genes	8 78	74 53	96 10	<b>1</b> 54	<b>2</b> <b>1</b>	<b>2</b> 12
no normalize						
first ranking	63 64	86 36	100 99	100 99	4 1	4 1
rev-del order	27 28	73 19	100 99	100 99	2 1	2 1
last two genes	12 41	56 97	82 96	97 82	<b>2</b> <b>1</b>	<b>2</b> <b>1</b>

Table 16: The number of patients in positive(+) class and negative(-) class of the training and testing datasets.

Data set												
	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)
train	23	17	25	15	24	16	19	21	17	23	23	17
test	626	374	686	314	608	392	528	472	560	440	584	416

Table 17: The (training/testing) classification accuracy rate for three different normalization cases, the first column is the number of genes still in the training set. The training set size is 40 examples, and the test set size is 1000 examples. Decision function  $x_1^2 + x_2^2 = c$ . Kernel: two degree polynomial.

# gene selected	accuracy rate					
	Normalization: gene $\rightarrow$ subjec					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.570	1/0.603	1/0.532	1/0.513	1/0.512	1/0.541
50	1/0.557	1/0.591	1/0.560	1/0.517	1/0.526	1/0.542
25	1/0.537	1/0.551	1/0.556	1/0.475	1/0.471	1/0.543
12	1/0.512	1/0.545	1/0.544	1/0.488	1/0.495	1/0.501
6	1/0.516	1/0.525	1/0.561	1/0.506	1/0.485	1/0.518
5	1/0.499	1/0.521	1/0.579	1/0.511	1/0.487	1/0.504
4	1/0.509	1/0.553	0.875/0.633	1/0.501	1/0.479	1/0.524
3	0.8/0.575	0.95/0.538	0.85/0.646	0.925/0.494	0.8/0.462	0.925/0.553
2	0.7/0.59	0.825/0.544	0.725/0.699	0.825/0.49	0.725/0.475	0.8/0.502
1	0.625/0.604	0.65/0.56	0.675/0.665	0.75/0.496	0.65/0.45	0.575/0.584

# gene selected	accuracy rate					
	Normalization: gene					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.551	1/0.602	1/0.533	1/0.523	1/0.514	1/0.534
50	1/0.527	1/0.578	1/0.547	1/0.504	1/0.53	1/0.540
25	1/0.550	1/0.549	1/0.558	1/0.51	1/0.536	1/0.530
12	1/0.521	1/0.529	1/0.517	1/0.541	1/0.575	1/0.512
6	1/0.507	1/0.539	1/0.468	1/0.522	1/0.625	1/0.503
5	1/0.504	1/0.519	1/0.463	1/0.522	1/0.641	1/0.513
4	1/0.480	1/0.518	0.925/0.506	0.925/0.511	1/0.645	1/0.542
3	0.825/0.479	0.875/0.546	0.8/0.509	0.75/0.471	0.95/0.654	0.9/0.519
2	0.65/0.558	0.775/0.587	0.625/0.57	0.675/0.476	0.9/0.646	0.775/0.502
1	0.575/0.62	0.6/0.579	0.6/0.608	0.575/0.496	0.9/0.7	0.575/0.584

# gene selected	accuracy rate					
	Normalization: not normalizing data					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.542	0.616	1/0.518	1/0.502	1/0.599	1/0.670
50	1/0.530	0.602	1/0.525	1/0.519	1/0.631	1/0.695
25	1/0.527	0.573	1/0.532	1/0.515	1/0.670	1/0.693
12	1/0.543	0.553	1/0.519	1/0.501	1/0.751	1/0.756
6	1/0.511	0.550	1/0.495	1/0.510	1/0.801	1/0.811
5	1/0.509	0.554	1/0.510	1/0.474	1/0.783	1/0.828
4	0.825/0.526	0.578	0.975/0.487	1/0.519	1/0.639	1/0.940
3	0.725/0.564	0.595	0.85/0.511	0.95/0.511	0.85/0.492	1/0.938
2	0.625/0.594	0.528	0.775/0.53	0.75/0.516	0.7/0.502	0.8/0.69
1	0.6/0.625	0.561	0.625/0.556	0.65/0.518	0.675/0.521	0.75/0.717



Table 18: The (training/testing) classification accuracy rate for three different normalization cases, the first column is the number of genes still in the training set. The training set size is 40 examples, and the test set size is 1000 examples. Decision function  $x_1^2 + x_2^2 = c$ . Kernel: RBF.

# gene selected	accuracy rate					
	Normalization: gene $\rightarrow$ subjec					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.574	1/0.629	1/0.579	1/0.527	1/0.477	1/0.632
50	1/0.626	1/0.673	1/0.608	1/0.526	1/0.608	1/0.686
25	1/0.604	1/0.649	1/0.594	1/0.514	1/0.688	1/0.734
12	1/0.581	1/0.594	1/0.554	1/0.520	1/0.709	1/0.760
6	1/0.539	1/0.597	1/0.566	1/0.508	1/0.725	1/0.856
5	1/0.550	1/0.579	1/0.558	1/0.499	1/0.713	1/0.844
4	0.95/0.528	0.95/0.575	0.975/0.57	1/0.512	1/0.780	1/0.833
3	0.925/0.551	0.825/0.616	0.85/0.567	0.925/0.492	1/0.813	1/0.827
2	0.825/0.511	0.7/0.626	0.725/0.549	0.675/0.493	1/0.855	1/0.924
1	0.8/0.548	0.675/0.679	0.625/0.562	0.525/0.502	0.9/0.694	0.875/0.729

# gene selected	accuracy rate					
	Normalization: gene					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.446	1/0.591	1/0.554	1/0.515	1/0.483	1/0.636
50	1/0.419	1/0.668	1/0.608	1/0.513	1/0.626	1/0.659
25	1/0.519	1/0.622	1/0.594	1/0.514	1/0.729	1/0.725
12	1/0.507	1/0.573	1/0.550	1/0.537	1/0.723	1/0.783
6	1/0.515	1/0.568	1/0.548	1/0.594	1/0.724	1/0.814
5	0.975/0.515	1/0.554	1/0.542	1/0.608	1/0.717	1/0.837
4	0.9/0.509	0.975/0.559	0.9/0.55	0.975/0.675	1/0.745	1/0.840
3	0.8/0.569	0.85/0.601	0.85/0.557	0.925/0.684	1/0.772	0.95/0.645
2	0.775/0.584	0.75/0.579	0.7/0.566	0.9/0.72	1/0.871	0.925/0.666
1	0.7/0.565	0.725/0.598	0.625/0.609	0.9/0.757	0.9/0.707	0.875/0.733

# gene selected	accuracy rate					
	Normalization: not normalizing data					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.462	1/0.603	1/0.640	1/0.516	1/0.49	1/0.632
50	1/0.411	1/0.576	1/0.484	1/0.514	1/0.65	1/0.691
25	1/0.53	1/0.548	1/0.512	1/0.509	1/0.732	1/0.75
12	1/0.549	1/0.532	1/0.507	1/0.524	1/0.752	1/0.774
6	1/0.534	0.925/0.532	1/0.535	1/0.505	1/0.741	1/0.772
5	0.975/0.543	0.9/0.543	1/0.547	1/0.513	1/0.722	1/0.783
4	0.975/0.559	0.85/0.533	1/0.518	1/0.5	1/0.761	1/0.804
3	0.8/0.546	0.8/0.578	1/0.492	1/0.485	1/0.789	1/0.903
2	0.7/0.55	0.725/0.553	1/0.514	0.975/0.488	1/0.955	1/0.979
1	0.5/0.486	0.725/0.524	0.8/0.508	0.7/0.51	0.90.715	0.85/0.744



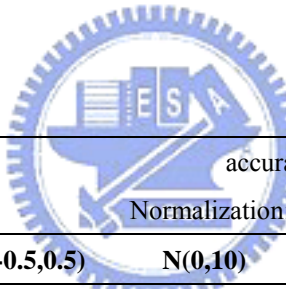
Table 19: The rankings of gene1 and gene2 in the first training (with 100 genes) and the reverse deleting by nonlinear SVM RFE with a polynomial kernel of degree 2 and RBF kernel with parameter  $\sigma = 3$  and the two genes stay to the end (Decision function  $x_1^2 + x_1x_2 + x_2^2 = c$ , six kinds of different data structures, and three different normalization cases ).

c	1	0.1	100	60	10	12
data	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
poly-2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2
Normalize (gene → subject)						
first ranking	65 50	58 64	65 12	31 46	74 92	81 94
rev-del order	69 70	80 81	53 62	68 15	65 90	85 92
last two genes	36 4	15 76	73 95	54 35	44 25	97 52
Normalize (gene)						
first ranking	79 67	48 53	69 38	15 57	53 90	88 97
rev-del order	84 83	25 76	47 35	5 14	35 83	64 97
last two genes	38 94	35 42	96 71	91 79	77 5	52 97
not normalizing						
first ranking	86 89	60 98	87 12	7 83	1 2	2 1
rev-del order	88 90	53 98	88 9	35 88	1 5	6 2
last two genes	85 58	15 94	194 61	20 85	1 55	33 2
rbf-3	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2	gene1 gene2
Normalize (gene → subject)						
first ranking	95 99	86 59	76 97	75 94	2 1	2 1
rev-del order	93 98	72 39	84 97	44 97	2 1	2 1
last two genes	71 20	66 48	98 49	26 33	2 1	2 1
Normalize (gene)						
first ranking	79 70	91 69	73 91	24 100	2 1	3 1
rev-del order	54 52	75 41	88 96	46 100	2 1	2 1
last two genes	8 85	44 97	40 6	36 62	2 1	2 1
not normalizing						
first ranking	81 74	88 83	100 99	100 99	2 1	2 1
rev-del order	45 50	91 92	100 99	100 99	2 1	1 2
last two genes	4 20	45 80	82 96	96 85	2 1	1 2

Table 20: The number of patients in positive(+) class and negative(-) class of the training and testing datasets.

Data set	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)
train	23	17	22	18	24	16	22	18	17	23	22	18
test	582	418	638	362	575	425	450	550	620	380	536	464

Table 21: The (training/testing) classification accuracy rate for three different normalization cases, the first column is the number of genes still in the training set. The training set size is 40 examples, and the test set size is 1000 examples. Decision function:  $x_1^2 + x_1x_2 + x_2^2 = c$ . Kernel: two degree polynomial.



# gene selected	accuracy rate					
	Normalization: gene $\rightarrow$ subject					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.534	1/0.527	1/0.548	1/0.485	1/0.486	1/0.506
50	1/0.539	1/0.511	1/0.531	1/0.491	1/0.443	1/0.509
25	1/0.514	1/0.535	1/0.541	1/0.510	1/0.471	1/0.503
12	1/0.488	1/0.496	1/0.524	1/0.510	1/0.475	1/0.491
6	1/0.509	1/0.513	1/0.491	1/0.519	1/0.491	1/0.500
5	1/0.492	1/0.489	1/0.503	1/0.482	1/0.512	1/0.516
4	1/0.503	1/0.488	0.875/0.497	0.9/0.502	1/0.513	1/0.510
3	0.975/0.489	0.95/0.486	0.725/0.516	0.9/0.512	0.825/0.499	0.8/0.518
2	0.775/0.482	0.75/0.451	0.65/0.533	0.65/0.5	0.8/0.464	0.725/0.505
1	0.575/0.582	0.725/0.529	0.625/0.561	0.6/0.484	0.70.441	0.6/0.528

# gene selected	accuracy rate					
	Normalization: gene					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.526	1/0.541	1/0.542	1/0.488	1/0.486	1/0.504
50	1/0.526	1/0.541	1/0.534	1/0.519	1/0.473	1/0.489
25	1/0.521	1/0.513	1/0.518	1/0.521	1/0.469	1/0.536
12	1/0.527	1/0.484	1/0.521	1/0.525	1/0.484	1/0.499
6	1/0.489	1/0.489	1/0.512	1/0.549	1/0.482	1/0.509
5	1/0.536	1/0.498	1/0.523	1/0.551	1/0.503	1/0.518
4	0.925/0.516	1/0.523	1/0.508	1/0.490	1/0.511	1/0.511
3	0.725/0.484	0.85/0.545	0.875/0.499	0.925/0.484	0.875/0.513	0.85/0.497
2	0.725/0.542	0.8/0.538	0.7/0.519	0.725/0.496	0.675/0.54	0.725/0.513
1	0.55/0.586	0.75/0.542	0.65/0.548	0.55/0.46	0.575/0.38	0.625/0.521

# gene selected	accuracy rate					
	Normalization: not normalizing data					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.518	1/0.551	1/0.533	1/0.476	1/0.567	1/0.669
50	1/0.509	1/0.553	1/0.507	1/0.477	1/0.603	1/0.673
25	1/0.499	1/0.521	1/0.545	1/0.510	1/0.638	1/0.689
12	1/0.518	1/0.512	1/0.562	1/0.508	1/0.725	1/0.706
6	1/0.526	1/0.520	1/0.524	1/0.508	1/0.795	1/0.729
5	1/0.506	1/0.529	1/0.524	1/0.483	1/0.822	1/0.653
4	1/0.501	1/0.499	0.975/0.493	1/0.470	1/0.620	1/0.670
3	0.9/0.479	0.9/0.546	0.775/0.503	0.9/0.48	0.90.651	0.95/0.68
2	0.75/0.493	0.8/0.546	0.775/0.514	0.675/0.486	0.875/0.675	0.95/0.702
1	0.575/0.582	0.75/0.536	0.6/0.575	0.55/0.45	0.8/0.697	0.675/0.484

Table 22: The (training/testing) classification accuracy rate for three different normalization cases, the first column is the number of genes still in the training set. The training set size is 40 examples, and the test set size is 1000 examples. Decision function:  $x_1^2 + x_1x_2 + x_2^2 = c$ . Kernel: RBF.

# gene selected	accuracy rate					
	Normalization: gene $\rightarrow$ subjec					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.531	1/0.516	1/0.554	1/0.502	1/0.542	1/0.601
50	1/0.582	1/0.560	1/0.577	1/0.523	1/0.400	1/0.647
25	1/0.564	1/0.519	1/0.571	1/0.499	1/0.590	1/0.698
12	1/0.548	1/0.503	1/0.546	1/0.516	1/0.627	1/0.757
6	1/0.528	1/0.511	1/0.544	1/0.510	1/0.740	1/0.836
5	0.975/0.512	0.975/0.523	0.975/0.541	1/0.502	1/0.731	1/0.873
4	0.975/0.51	0.975/0.495	0.925/0.526	0.95/0.504	1/0.760	1/0.875
3	0.9/0.527	0.9/0.518	0.9/0.533	0.925/0.484	1/0.820	1/0.886
2	0.775/0.482	0.8/0.552	0.8/0.528	0.775/0.496	1/0.834	0.975/0.906
1	0.65/0.578	0.725/0.531	0.65/0.566	0.55/0.45	0.85/0.625	0.9/0.716

# gene selected	accuracy rate					
	Normalization: gene					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.473	1/0.536	1/0.555	1/0.511	1/0.487	1/0.613
50	1/0.537	1/0.532	1/0.575	1/0.505	1/0.402	1/0.646
25	1/0.518	1/0.528	1/0.577	1/0.486	1/0.566	1/0.693
12	1/0.515	1/0.521	1/0.554	1/0.498	1/0.643	1/0.761
6	1/0.555	1/0.509	1/0.521	1/0.499	1/0.739	1/0.767
5	1/0.559	1/0.522	0.975/0.533	1/0.513	1/0.755	1/0.808
4	0.95/0.55	0.95/0.531	1/0.534	0.975/0.51	1/0.832	1/0.816
3	0.85/0.555	0.875/0.507	0.925/0.518	0.85/0.487	1/0.833	1/0.95
2	0.8/0.568	0.75/0.511	0.825/0.51	0.75/0.495	1/0.844	1/0.948
1	0.75/0.543	0.65/0.508	0.625/0.564	0.725/0.495	0.85/0.649	0.875/0.723

# gene selected	accuracy rate					
	Normalization: not normalizing data					
data set	<b>N(0,1)</b>	<b>Uni(-0.5,0.5)</b>	<b>N(0,10)</b>	<b>Uni(-10,10)</b>	<b>N(2,1)</b>	<b>Uni(1,3)</b>
100	1/0.475	1/0.531	1/0.064	1/0.525	1/0.488	1/0.611
50	1/0.521	1/0.544	1/0.5	1/0.511	1/0.418	1/0.66
25	1/0.536	1/0.522	1/0.515	1/0.513	1/0.613	1/0.721
12	1/0.521	1/0.528	1/0.532	1/0.497	1/0.658	1/0.767
6	1/0.496	0.925/0.523	1/0.518	1/0.529	1/0.863	1/0.865
5	1/0.481	0.8/0.569	1/0.533	1/0.536	1/0.853	1/0.85
4	0.925/0.521	0.8/0.551	1/0.548	1/0.51	1/0.912	1/0.932
3	0.775/0.516	0.75/0.575	1/0.527	1/0.486	1/0.928	1/0.951
2	0.75/0.536	0.75/0.568	1/0.535	1/0.497	1/0.942	0.975/0.974
1	0.575/0.582	0.625/0.534	1/0.494	0.725/0.508	0.85/0.658	0.8/0.762

Table 23: Results obtained in [19], Weston *et al.* The times of the correct features were selected over 30 random trials for various training set sizes using nonlinear SVMs with a polynomial kernel of degree 2.

Training set size	10	20	30	40	50	75	100
Times	2/30	19/30	27/30	28/30	26/30	30/30	28/30

Table 24: Results obtained in [19], Weston *et al.* The times of the correct feature was selected over 30 random trials for various training set sizes using our criteria in sec.4.2 (ii). The second row means the times of the correct feature was selected in the reverse order 2 training, and the third row means the times of the correct feature was selected in the last training.

Training set size	10	20	30	40	50	75	100
Times (top two)	7/30	25/30	28/30	28/30	30/30	29/30	30/30
Times (top one)	4/30	20/30	22/30	22/30	23/30	24/30	26/30

Table 25: Results obtained in [19]. The table shows the average test error rate and standard deviation on a test set of 500 examples over 30 random trials using two different feature selection methods. We plot them in Figure 2.

Training set size	Nonlinear SVMs (kernel:poly-2)	Map data to higher-dim space (top two)	Map data to higher-dim space (top one)
10	0.46007±0.09707	0.40913±0.13741	0.42127±0.14760
20	0.19707±0.16651	0.16120±0.13818	0.14313±0.14520
30	0.13780±0.12436	0.11500±0.05338	0.08673±0.04508
40	0.10233±0.05609	0.08907±0.05214	0.08580±0.04334
50	0.10073±0.09191	0.08660±0.03668	0.07773±0.04640
75	0.07073±0.03637	0.07313±0.04836	0.07680±0.04870
100	0.06213±0.03211	0.05560±0.03456	0.06093±0.02692

Figure 2: Results obtained in [19]. The x-axis is the training set size, and the y-axis is the average test error rate on a test set of 500 examples over 30 random trials.

