Interaction-Based Learning for High-Dimensional Data with Continuous Predictors

Chien-Hsun Huang

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2014

ABSTRACT

Interaction-Based Learning for High-Dimensional Data with Continuous Predictors

Chien-Hsun Huang

High-dimensional data, such as that relating to gene expression in microarray experiments, may contain substantial amount of useful information to be explored. However, the information, relevant variables and their joint interactions are usually diluted by noise due to a large number of non-informative variables. Consequently, variable selection plays a pivotal role for learning in high dimensional problems. Most of the traditional feature selection methods, such as Pearson's correlation between response and predictors, stepwise linear regressions and LASSO are among the popular linear methods. These methods are effective in identifying linear marginal effects but are limited in detecting non-linear or higher order interaction effects. It is well known that epistasis (gene - gene interactions) may play an important role in gene expression where unknown functional forms are difficult to identify. In this thesis, we propose a novel nonparametric measure to first screen and do feature selection based on information from nearest neighborhoods. The method is inspired by Lo and Zheng's earlier work (2002) on detecting interactions for discrete predictors. We apply

a backward elimination algorithm based on this measure which leads to the identification of many influential clusters of variables. Those identified groups of variables can capture both marginal and interactive effects. Second, each identified cluster has the potential to perform predictions and classifications more accurately. We also study procedures how to combine these groups of individual classifiers to form a final predictor. Through simulation and real data analysis, the proposed measure is capable of identifying important variable sets and patterns including higher-order interaction sets. The proposed procedure outperforms existing methods in three different microarray datasets. Moreover, the nonparametric measure is quite flexible and can be easily extended and applied to other areas of high-dimensional data and studies.

Contents

List of Tables

List of Figures

Acknowledgments

My deepest gratitude is to my advisor, Professor Shaw-Hwa Lo, for his advise and constant encourangment over the years and making this work possible. Discussions with Professor Lo always lead to a great sense of clarity and inspiration of my thoughts on scientific problems. I really thank him for having been always patient, supportive and available to me. I have been truely lucky to have him as my advisor.

I would like to thank Professor Tian Zheng, Professor Peter Orbanz, Professor Iuliana Ionita-Laza and Professor Pei Wang for kindly agreeing to serve on my committee and for their valuable feedback to improve my work. My gratitude goes to all my friends for their constant encouragement and accompany these years.

Last but most importantly, I want to express my profound gratitude to my parents and brothers for their supports and unconditional loves. I also owe my deepest thanks to my wife Nien-Tzu; Without her sacrifice and companionship, it would have been impossible to pursue my PhD. And thank you for bringing me such a cute boy, Edwin. His coming is one of the best things in my life

Dedicated to my family

Chapter 1

Introduction

1.1 Background

Developments in technology have led to an increasing amount of data available in many scientific fields. The relevant researchers are typically required to carry out their analysis in this data-rich environment. High-dimensional data, such as genome-wide human SNP array and DNA microarray, may contain a vast amount of useful information to be explored. However, the true relevant and influential features are always concealed by a huge number of noisy and irrelevant features. Furthermore, identifying epistasis and high order interactions also creates a new challenge for scientific works. Large numbers of variables and the small number of observations make it more difficult to identify true signals. To deal with these problems, many variable selection methods are proposed and developed. The merits of carrying out feature selection are several fold. First, it can avoid overfitting problems and reduce the noise to improve model performance. Second, it can reduce the computational complexity by fitting models with a subset of important variables. Third, it can improve the interpretability of the models.

Most existing feature selection methods are based on the assumption that a linear

relationship holds between response and explanatory variables. For example, the measure of similarity, *Pearson's correlation coefficient* defined as the following:

$$cor(x, y) = \frac{E[(x - E[x])(y - E[y])]}{\sqrt{Var[x]Var[y]}} \ ,$$

is one of the most commonly used measure to screen the significant variables. It assesses the relevance between variables and response one at a time. To carry out the screening, the absolute correlation scores of all variables are first calculated and the low-scoring variables treated as unrelated with the response are removed. The main advantage of this method is that it can be applied in order to screen a large number of variables easily and quickly. However, there are some limitations. First, it can only evaluate the association between one predictor and response one at a time. Second, it can evaluate linear effects only. Third, outliers may have great influence on it. Fourth, it fails to detect interactive effects when interactions of several variables play an important role and the marginal effect is low.

Stepwise regression in a linear regression model is another popular method commonly used in feature selection. The forward stepwise selection is a greedy algorithm that produces a nested sequence of models that adds one new variable at a time based on indices such as deviance, Aikaike's Information Criterion (AIC) or Bayesian Information Criterion (BIC). For high dimensional data, the computation is easier and the solution is stable. As for backward selection, on the contrary, it sequentially deletes the predictor that has least impact. However, it cannot be used for data with more predictors than observations that make it infeasible in current high dimensional environment.

Another linear approach of variable selection is based on regularization method such as LASSO (i.e. least absolute shrinkage and selection operator) (Tibshirani, 1996). The LASSO (L-1 regularization) estimates based on optimizing the loss func-

tion

$$(\hat{\alpha}, \hat{\beta}) = \arg\min \left\{ \sum_{i=1}^{N} \left(y_i - \alpha - \sum_j \beta_j x_{ij} \right)^2 \right\}$$
$$s.t. \quad \sum_j |\beta_j| \leq t$$

where $t \geq 0$ is a tuning parameters. As t goes to infinity, the solution will be the same as linear regression estimates. Letting t be sufficiently small will automatically eliminate irrelevant variables shrinking their coefficient to zero. The LASSO does not pre-select a subset of variables but carries out a continuous shrinking operation. Unlike the backward stepwise selection method, it avoids the singularity issue by solving an optimization problem.

Classification is a specialty for supervised learning problems. In real world, the size of a dataset is so large that learning might not work as well before removing irrelevant variables. Therefore, many different methods are proposed to perform variable selection in the literatures. These methods can be organized into three categories (Saeys et al, 2007).

Filtering method includes Pearson's correlation, t-test, signal-to-noise ratio (Gloub et al, 1999), information gain (Ben-Bassat, 1982), wilcoxon rank sum (Dettling & Buhlmann, 2003), Significance Analysis of Microarray (SAM) (Tusher et al, 2001) and Threshold Number of Misclassification (TNoM) scores (Ben-Dor et al, 2000). These methods are used to compute a score for each variable and those of highly relevant variables are selected. Although boosting the advantage of computational efficiency, these measures are independent of classifiers and ignore the dependencies among predictors.

Wrapper methods search the best subsets by greedy algorithms. For instance, there are sequential forward/backward selection (Kittler, 1978) and supervised clustering of genes (Dettling & Buhlmann, 2003), which defines a score function to find the most informative groups of subset. Another popular wrapper method, genetic

algorithm (Holldan, 1975), is used to search large spaces with little known in prior. In general, the wrapper methods are more computationally intensive.

The *embedded* methods search the optimal subset of predictors as part of model construction. The selected predictors interact with used classifiers well but may not work well with other classifiers. Many methods have been developed based on existing classifiers, such as the variable importance that measure the contribution each variable makes in randomforest and classification and regression tree (CART). SVM recursive feature elimination (SVM-RFE), which ranks the predictors based on the weight of coefficient and eliminates the one with lowest weight (Guyon et al, 2002); Recursive SVM (R-SVM) evaluates the contribution of predictors by the weight of coefficient and the difference of two class mean (Zhang et al, 2006).

Besides the variable selection methods, some dimension reduction techniques are introduced to perform a linear mapping of data to a lower dimensional space, such as principal component analysis (PCA) and singular value decomposition (SVD) (Wall, 2001). Both of the methods are used to find a small set of orthogonal linear combinations of the original predictors that are optimal at capturing the underlying variance of the data. However, they only measure the variability of the predictors without considering the contribution of variables towards the responses. In classification problem, ignoring the relation to the response may make the key components suffer from losing the information of the classes and consequently lead to inaccuracy in classification.

1.2 Organization of the Thesis

In this thesis, we proposed a novel influence measure and its applications. The remainder of this thesis is organized as follows. In Chapter 2 we discuss related studies and propose the nonparametric variable selection measure based on neighborhood information to identify influential variables and their interactions. In Chapter 3 we discuss the asymptotic property of proposed influence measure. In Chapter 4 we

propose backward elimination algorithm and discuss the details of this algorithm. In Chapter 5, an interaction-based classification framework is proposed to perform classification as follows. First, we perform pairwise-screening the variables by proposed measure and select high return frequency variables. Second, a backward elimination algorithm is applied to the selected variables to form informative building blocks. Third, filtering out non-overlapped building blocks and aggregate them to perform prediction by a classifier aggregation method. Chapter 6 we provide a discussion and concluding remarks.

Chapter 2

Nonparametric Influence Measure - I Score

In this chapter, we first introduce our basic tool, a novel influence measure I for continuous variables. The main properties of it will be outlined and discussed.

2.1 Relations with Previous Studies

Given a vector of predictors $\mathbf{X}=\{X_1, X_2, \ldots, X_m\}$ where m is the total number of continuous explanatory variables, we want to predict a real-value dependent variable Y, which is a vector of n observations. In general, the model between (Y, $\mathbf{X}$) can be formulated as

$$Y = f(\mathbf{X}) + \epsilon, \tag{2.1}$$

where ϵ is an error vector with each element ϵ_i as $N(0, \sigma^2)$, i=1,..., n and the function $f(.)$ can be any form. One special case of the model involves the assumption of function $f_L(.)$ to be linear additive relationship between Y and $\mathbf{X}$. If the number of observations n is greater than the number of variables m, the functional form of $\hat{f}_L(.)$

is estimated by least square methods. The relevant properties in the linear model are well documented.

To evaluate the adequacy of the fitted model under the linear assumption, the coefficient of multiple determination is commonly used and defined as follows:

$$R^2 \;=\; \frac{Var(\hat{f}_L(\mathbf{X}))}{Var(Y)} = \; cor^2(Y, \hat{f}_L(\mathbf{X})) = \; cor^2(Y, \hat{Y}) \tag{2.2}$$

where $\hat{Y}$=$\hat{f}_L(X)$. The R^2 is not merely a measure of the strength of the linear relationship but also gives the fraction of the variability of Y that is explained by $\mathbf{X} \in \mathbf{R}^m$ (or linear joint effect of $\mathbf{X}$ on Y). The coefficient of multiple correlation, R, can be treated as an index to examine the strength of association between response (Y) and estimated value ($\hat{Y}$). In the multiple linear regression model, R is always between 0 and 1. $R = 1$ indicates the model is a perfect fit and higher positive value of R indicate Y and $\hat{Y}$ are close to each other.

Other than the linear additive assumption, Doksum and Samarov (1995) proposed the nonparametric coefficient of determination similar to (2.2) to evaluate the importance of a subset of covariates where the estimates ($\hat{f}(\mathbf{X}_i)$) are computed wtih "leave-one-out" kernel estimators:

$$\hat{f}(\mathbf{X}_i) \;=\; \frac{(n-1)^{-1}\sum_{j\neq i}^{n} Y_i K_h(||\mathbf{X}_j - \mathbf{X}_i||)}{(n-1)^{-1}\sum_{j\neq i}^{n} K_h(||\mathbf{X}_j - \mathbf{X}_i||)}, \quad where\ i = 1, ..., n \tag{2.3}$$

where $K_h(||\mathbf{X}_j - \mathbf{X}_i||)$ is the kernel density function with bandwidth h computed based on the distance between i^{th} and j^{th} observations. However, the kernel estimators using a variable bandwidth may be less efficient since it is computational expensive to optimally tune the bandwidth by cross-validation methods. This method might also ignore the local patterns since the effective number of observations that are used to estimate $\hat{f}(\mathbf{X}_i)$ varies with the distribution of $\mathbf{X}$. In the high dimension problem, instead of screening the important variable sets, the method with varied bandwidth will increase the computational burden and hence is not efficient. Furthermore, the defined nonparametric coefficient of determination based on correlation square may

have chances to select a subset lacking of predictive power when unsimilar responses are prone to cluster together in such a selected subset. To ensure the predictive power, only those with positive correlation are necessary to be considered.

On the other hand, when the explanatory variables $\mathbf{X}$ are discrete, Chernoff, Lo and Zheng (2009) proposed the Partition Retention method to detect both marginal and high-order interaction effects based on Lo and Zheng's earlier work (2002). Assume that $\{X_j, j = 1, ..., m\}$ and that all the explanatory variables only take on the values 0 and 1. Then there will be 2^m possible partitions for each set of m explanatory variables. The n observations are partitioned into the 2^m partition elements or cells. They define the normalized influence score as:

$$I = \frac{\Sigma_{k=1}^{2^m} n_k^2 (\hat{Y}_k - \bar{Y})^2}{n\sigma_Y^2}, \tag{2.4}$$

where $\hat{Y}_k$, the estimated value, is the average of the n_k observations on Y falling in the k^{th} partition cell, $\bar{Y}$ is the grand mean of Y and σ_Y^2 is the variance of Y. It can be shown (Chernoff, 2009) that under the null hypothesis that there is no influence among the $\mathbf{X}$, the asymptotic distribution of I is very close to that of a weighted average of independent chi-squares with one degree of freedom each and their weights are proportional to the n_k (i.e. $\Sigma_{k=1}^{2^m} \frac{n_k}{n} \chi_1^2$).

In fact, the score I and R^2 are similar. If the explanatory variables are continuous, each observation forms an individual cell by itself (i.e. n_k=1). Therefore, there will be n cells and the influence score is:

$$\begin{aligned} I &= \frac{\Sigma_{k=1}^{n} (\hat{Y}_k - \bar{Y})^2}{n\sigma_Y^2} \\ &= \frac{\frac{1}{n}\Sigma_{k=1}^{n} (\hat{Y}_k - \bar{Y})^2}{\sigma_Y^2} \\ &= \frac{Var(\hat{f}(\mathbf{X}))}{Var(Y)} \end{aligned} \tag{2.5}$$

If we further assume that $\hat{Y}_k$ in (2.5) is estimated by linear function $\hat{f}_L(.)$, the two scores I and R^2 in (2.2) are nearly equivalent.

2.2 Nonparametric Influence Measure

When the explanatory variables are continuous, we propose a nonparametric method based on nearest neighborhood information that accommodates a flexible form of the regression curve or specific patterns in the space. Let $\mathbf{X}_i = (X_{i1}, X_{i2}, \ldots, X_{im})$ as i^{th} $\mathbf{X}$-observation in m dimensional Euclidean space. Denote the distance between i^{th} and j^{th} point as:

$$d(\mathbf{X}_i, \mathbf{X}_j) = \Sigma_{r=1}^{m}(X_{ir} - X_{jr})^2.$$

For each i^{th} $\mathbf{X}$-observation, the k^{th} nearest neighbors of $\mathbf{X}_i$ are denoted by $\{\mathbf{X}_{i(1)}$,$\mathbf{X}_{i(2)}$, ...,$\mathbf{X}_{i(k)}\}$ and their responses are denoted by $\{Y_{i(1)}$,$Y_{i(2)}$,...,$Y_{i(k)}\}$. We also define a generalized weighting matrix

$$W = \{w_{ij},\ w_{ij} \geq 0, \forall i, j\ \},$$

where the weight w_{ij} denotes the effects of observation j on the i^{th} observation. The weight $w_{i,i(s)}$ assigned to the s^{th} nearest neighbor is determined by a monotonic decreasing function $K(\mathbf{X}_i, \mathbf{X}_{i(s)})$ as s increases.

Let $\tilde{Y}_i$ be the estimate of Y_i by the weighted average of the remaining n-1 observations as follows:

$$\tilde{Y}_i = \hat{E}(Y|\mathbf{X} = \mathbf{X}_i) = \Sigma_{s=1}^{n-1} w_{i,i(s)} Y_{i(s)} \tag{2.6}$$

where

$$\begin{aligned} w_{i,i(s)} &= \begin{cases} 0 & s = 0 \\ \dfrac{K(\mathbf{X}_i, \mathbf{X}_{i(s)})}{\Sigma_{s=1}^{n-1} K(\mathbf{X}_i, \mathbf{X}_{i(s)})} & otherwise, \end{cases} \\ \sum_{s=1}^{n-1} w_{i,i(s)} &= 1 \quad \forall i = 1, 2, \ldots, n, \ and \\ s &= 0 \ indicates \ the \ i^{th} \ point \ itself. \end{aligned}$$

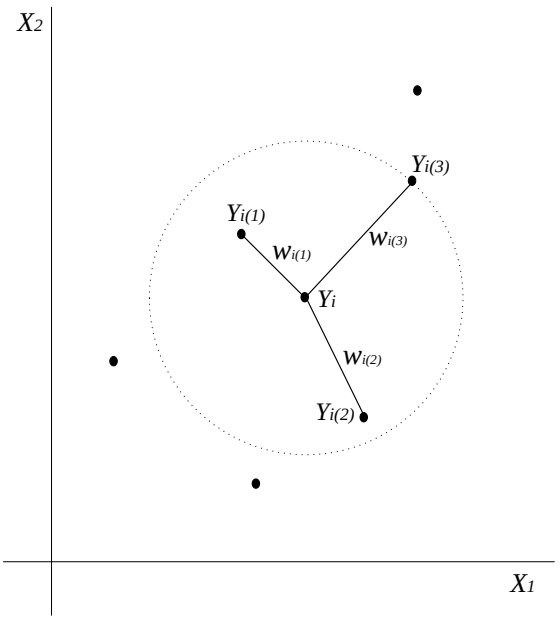

Figure 2.1: 3-NN rule in 2 dimensional spaces

The weight matrix W is determined by the data distributed in the space. The diagonal elements of the matrix are 0 and the elements of each row are normalized so that sum of the weights from each row is 1. In addition, W may not be symmetric, that is $w_{ij} \neq w_{ji}$ for $i \neq j$.

Figure 2.1 shows an illustration of 3NN rule of two dimensional spaces $\{X_1, X_2\}$. For observation i, the 3 nearest points located inside the circle are $\{Y_{i(1)}, Y_{i(2)}, Y_{i(3)}\}$ with the corresponding weights $\{w_{i(1)}, w_{i(2)}, w_{i(3)}\}$. The estimated value $\tilde{Y_i}$ of Y_i is the weighted sum of them $\Sigma_{s=1}^{3} w_{i,i(s)} Y_{i(s)}$. Methods of assigning weights are discussed in the next section.

For each observation Y_i, $i = \{1, \ldots, n\}$, its corresponding estimate $\tilde{Y_i}$ is obtained by (2.6). Based on similar idea to the general marginal correlation method

($i.e.$ $cor(Y, X_i)$) which is applied to measure the effect of X_i on Y, we propose the following correlation coefficient as a measure of joint influence of **X** on Y.

$$I = cor(Y, \tilde{Y}) \tag{2.7}$$

where $\tilde{Y}$ is a vector of estimate on Y.

2.3 Basic Properties of Influence Measure

The influence score I has a number of desirable properties:

Property 1. *I score takes values between -1 and 1*

Under the linear assumption, the definition of I is similar to coefficient of multiple correlation R, which is never a negative value since it is the fraction of variance explained over total variance. However, the nonparametric I score is not guaranteed to be positive since the score depends on the underlying data distribution and the selected parameters.

Property 2. *Only variable set with high positive values of I have predictive power.*

Higher positive value of I indicates a stronger influence and predictive power of the joint **X**. That means similar responses of Y cluster together in the joint variable space. On the contrary, the score that is close to zero or even negative suggests a weak prediction.

Property 3. *I score has the ability to detect both linear and non-linear patterns.*

Property 4. *I score is capable to identify local and global patterns.*

I is able to capture both local or global patterns by adjusting the weight of nearest neighbors. If the data shows obvious local pattern, we can choose small k to catch the pattern. If there are strong global patterns, both large and small k are able to capture influential variables.

Property 5. *I is easily computed*

The score is easy and fast to compute, it is efficient to preliminarily screen potential

marginal or interaction effects in a high dimension data set.

Property 6. *Important information remains if noisy variables are removed*
If X_1 is not important, the I score based on the joint set $\mathbf{X}(1) = \mathbf{X} \setminus \{X_1\}$ will remain strong.

2.4 Inverse Distance Weighted Function

In general, the k nearest neighborhood method takes an average of the k nearest points with equal weight to the estimation. All of these k observations make the same contributions to estimate the target. If the data is distributed in a sparse region, some of the k nearest points may be located far away from the target and are irrelevant to it. It does not make much sense that these points have the same impact as the first few nearest neighbors. Therefore, we will discuss a number of weighting functions in this section:

Inverse Rank Weights (I_r)
This weight function assigns weights based on their inverse rank. More specifically, the estimated value $\tilde{Y_i}$ of Y_i is calculated by the function:

$$\tilde{Y_i} = \Sigma_{s=1}^{k}(Y_{i(s)} \cdot \frac{k+1-s}{k(k+1)/2}) \tag{2.8}$$

where $Y_{i(s)}$ is the response of the s^{th} neighbor corresponding to observation i. For example if k =3, the weight is assigned as $(\frac{3}{6}, \frac{2}{6}, \frac{1}{6})$ to the three neighbors such that the numerator is rank of the k nearest neighbors and the denominator is sum of all ranks.

Inverse Distance Weights (I_d)
This weight function assigns weights proportional to the inverse distance of the target and $\tilde{Y_i}$, and is defined as

$$\tilde{Y_i} = \frac{\Sigma_{s=1}^{k} w_s Y_{i(s)}}{\Sigma_{s=1}^{k} w_s} \tag{2.9}$$

where

$$w_s = \frac{1}{d(\mathbf{X}_i, \mathbf{X}_{i(s)})} \tag{2.10}$$

where $d(X_i, X_{i(s)})$ is the Euclidean distance of the i^{th} observation and its s^{th} nearest neighbor.

Nadaraya-Watson Kernel-Weighted Average (I_k)

The weight function uses the *Epanechnikov* quadratic kernel with certain window size function $h_\lambda(x)$:

$$K_\lambda(\mathbf{X}_i, \mathbf{X}_{i(s)}) = D(\frac{|\mathbf{X}_{i(s)} - \mathbf{X}_i|}{h_\lambda(x)}) \tag{2.11}$$

$$D(t) = \begin{cases} \frac{3}{4}(1-t^2) & |t| \leq 1; \\ 0 & otherwise \end{cases}$$

$h_\lambda(x)$ is set as a constant if only fixed window size is considered. The estimated value $\tilde{Y}_i$ is calculated as (2.6). For k nearest neighborhoods, $h_\lambda(x) = |\mathbf{X}_i - \mathbf{X}_{i(k)}|$ where $\mathbf{X}_{i(k)}$ is the k^{th} nearest neighbors of the i^{th} observation.

2.5 Simulation Studies of Proposed Influence Measure

In this section, a few one-dimension and two-dimension functional relationships are used to illustrate the capability to catch the influential variable by the I score with varied weight functions. Given that the influential predictor X is from uniform distribution with an appropriate ranges and the random noise (ϵ) is chosen from standard normal distribution with N=200 observations. In table 2.1, we compare marginal correlation score (i.e. Pearson's correlation) and influence scores by three different kernels with varied k=3, 5, 10. The scatterplots of simulated data and their underlying functions are plotted in figure 2.2.

Table 2.1: Correlation and influence measures in one dimensional functions

N=200	**Underlying Model (Y)**				
	$X+\epsilon$	$X^2+\epsilon$	$e^X+\epsilon$	$log(X)+\epsilon$	$cosX+\epsilon$
cor(y,x)	0.9363	-0.0148	0.5870	0.5596	0.0008
$k=3$					
$I_r(x)$	0.9172	0.9502	0.9398	0.5684	0.5686
$I_d(x)$	0.8969	0.9389	0.9282	0.5237	0.5299
$I_k(x)$	0.9056	0.9430	0.9316	0.5216	0.5447
$k=5$					
$I_r(x)$	0.9232	0.9464	0.9552	0.6118	0.5918
$I_d(x)$	0.8993	0.9309	0.9411	0.5346	0.5386
$I_k(x)$	0.9198	0.9437	0.9529	0.5914	0.5815
$k=10$					
$I_r(x)$	0.9303	0.9511	0.9550	0.6563	0.6349
$I_d(x)$	0.9015	0.9324	0.9402	0.5434	0.5482
$I_k(x)$	0.9303	0.9512	0.9550	0.6552	0.6284

I_r: Score with inverse rank weight
I_d: Score with inverse distance weight
I_k: Score with *Epanechnikov* kernel weight

The marginal correlation score may capture the influences of linear relations, exponential and log functional effects. For the proposed influence scores, the three higher influence scores demonstrate the capability of detecting the effects of X on Y under both linear and nonlinear situations. In addition to that, the proposed method is able to detect signals based on other nonlinear functional forms. Furthermore, in general, the scores increase as k increases since simulated functions exhibit global patterns. In reality, there might be specific local patterns for which moderate k is preferred. In addition, we also observed that these influence scores with different kernels are quite similar, while I_r tends to be slightly higher than the other two.

It is expected, in practice, that a wide array of nonlinear associations may exist besides linearity. The marginal correlation score or other methods based on linear assumptions will fail to detect strong nonlinear association and other joint effects.

It is entirely possible that $cor(Y, X_i) = 0$ or near zero while X_i and Y are functionally dependent. The key advantage of proposed influence measure I is its ability to identify influences in arbitrarily specific global and local structures. Furthermore, it automatically considers interaction effects among predictors when calculating the scores with neighborhood information.

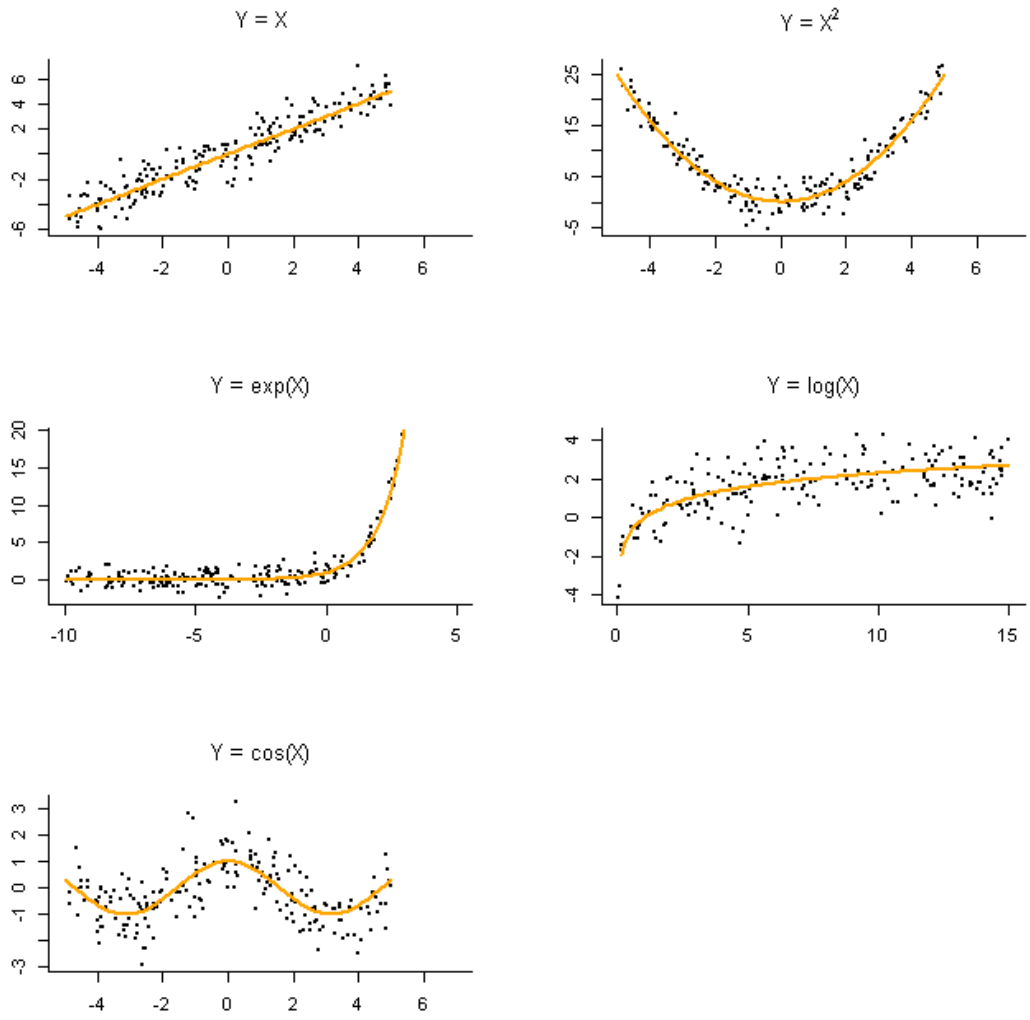

Figure 2.2: Scatterplot and underlying curve (orange) of different functional forms

Table 2.2 lists the pearson's correlation of each individual variable and proposed

influence scores with two variables in different models. Assume that the two variables (X_1, X_2) have certain effects on response (Y) in different functional forms and the number of observation $N = 200$. Beside the log functional form that X_1 and X_2 are generated from $U(0, 10)$, all the models have X_1, X_2 and random noise ϵ from standard normal distribution $N(0, 1)$.

Table 2.2: Correlations and influence measures in two dimensional functions

$N = 200$	**Underlying Model (Y)**					
	$X_1 + X_2$	X_1X_2	$X_1^2 + X_2^2$	$e^{X_1X_2}$	$log(X_1X_2)$	$sin(X_1X_2) + cos(X_1X_2)$
cor(y,x_1)	0.5900	0.0848	-0.1176	-0.1513	0.4100	0.1512
cor(y,x_2)	0.5537	-0.0664	-0.0082	-0.1417	0.5162	-0.0152
$k = 3$						
$I_r(x_1, x_2)$	0.7431	0.5863	0.8064	0.7224	0.6686	0.4775
$I_d(x_1, x_2)$	0.7386	0.5777	0.8001	0.7321	0.6309	0.4737
$I_k(x_1, x_2)$	0.7126	0.5535	0.7876	0.7429	0.6233	0.4459
$k = 5$						
$I_r(x_1, x_2)$	0.7615	0.6029	0.8167	0.6909	0.6966	0.4966
$I_d(x_1, x_2)$	0.7532	0.5889	0.8084	0.7218	0.6454	0.4902
$I_k(x_1, x_2)$	0.7529	0.5952	0.8157	0.6901	0.6809	0.4874
$k = 10$						
$I_r(x_1, x_2)$	0.7771	0.6246	0.8214	0.6723	0.7061	0.5159
$I_d(x_1, x_2)$	0.7649	0.6069	0.8117	0.7218	0.7209	0.5012
$I_k(x_1, x_2)$	0.7764	0.6211	0.8243	0.6901	0.6795	0.5170

We observe that the magnitude of marginal correlation coefficients shows signals in both linear and log functional relationships, but fail to capture other effects like joint and high order. Applying proposed influence measure by including these two influential variables to high order or interactive functional form, these scores are all high enough, demonstrating the method's ability to identify potential joint effects of multiple variables. In addition, with different values of k, the scores among different weight functions perform similarly.

Figure 2.3 also shows two different functional plots with two predictors. In the

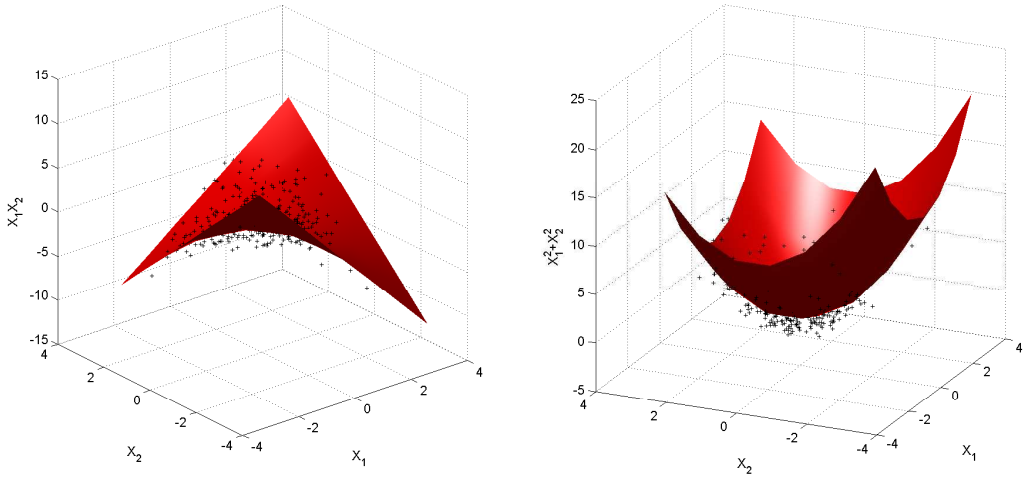

Figure 2.3: Scattplots and underlying surfaces (red) of two dimensional functions

first function, the response has joint effects from both X_1 and X_2, and the second function is a paraboloid function. Both of them are undetectable by linear screening methods without knowing the specific functional form. The proposed method takes advantage of neighborhood information that helps us identify many different nonlinear or interactive effects.

From these simulations, the proposed influence measures are able to spot the influential variable sets by recognizing the specific patterns between response and predictors. In high dimensional data sets, they provide novel and effective ways to screen the potential important variables. Compared with the Pearson's correlation, the proposed measure I is able to locate important variables under either linear or nonlinear relations. It also provides an efficient and flexible way to screen variables with higher order and interactive effects.

Chapter 3

Asymptotic Property of Measure I

In this part, we shall discuss the asymptotic properties of the proposed influence measure.

3.1 Asymptotic Distribution of I

Let $Y_1, Y_2, \ldots, Y_n$ be independent identically distributed with mean μ and variance σ^2. Without lose of generality, we can define the centered variates Z_i corresponding to the observed values $z_i = y_i$ - $\bar{y}$. Therefore

$$\begin{aligned} E[Z_i] &= E[y_i - \bar{y}] = 0 \\ E[Z_i^2] &= E[(y_i - \bar{y})^2] = E[y_i^2] - E[\bar{y}^2] = (\mu^2 + \sigma^2) - (\frac{\sigma^2}{n} + \mu^2) = (1 - \frac{1}{n})\sigma^2 \\ E[Z_i Z_j] &= E[(y_i - \bar{y})(y_j - \bar{y})] = E[y_i y_j] - E[y_i \bar{y}] - E[y_j \bar{y}] + E[\bar{y}^2] \\ &= \mu^2 - 2\frac{n\mu^2 + \sigma^2}{n} + (\frac{\sigma^2}{n} + \mu^2) = -\frac{\sigma^2}{n} \end{aligned}$$

Define $w_i = \{w_{i1}, w_{i2}, \ldots, w_{in}\}$ as the vector of weight between i^{th} subject and all the other $n-1$ observations where $w_{ii} = 0$ and $\Sigma_{j=1}^{n} w_{ij} = 1 \ \forall i \in \{1, \ldots, n\}$. Many different weighting methods have been discussed in section 2.4 to assign the vector of w_i. In the following, we provide the asymptotic property of proposed influence

measure. First, we introduce the theorem:

Theorem 3.1.(**Guyon, 1995**)

Consider $S \subset \mathbb{R}^d$ an infinite nonnecessarily regular lattice without accumulation points and D_n be an increasing sequence of finite subsets of S. Let W be a matrix of known bounded weights over S^2 such that

$$\{W = (w_{ij}),\ X_i, X_j \in S\} \quad w_{ii} = 0, w_{ij} = 0 \text{ if } ||X_i - X_j|| > R$$

Define the measure

$$\rho_n = \Sigma_{i \in D_n'} \Sigma_j w_{ij} Z_i Z_j$$

where $D_n^{'} = \{X_k \in D_n$ for all $X_\ell \in D_n$ that $w_{k\ell} \neq 0\}$.

In addition, for $c_n = \Sigma_{i \in D_n'} \Sigma_j (w_{ij}^2 + w_{ij} w_{ji})$ and $Var(Z_i) = \sigma^2$ for all i . If

(i) *The variables Z_i are centered, independent and $\exists \delta > 0$ that $||Z||_{2+\delta} = sup_i ||Z_i||_{2+\delta} < \infty$*

(ii) $\liminf_n c_n |D_n^{'}|^{-1} > 0$, *then*

$$(c_n)^{-1/2} (\Sigma_{D_n} Z_i^2)^{-1} \rho_n \xrightarrow{D} N(0,1) \tag{3.1}$$

Consider the data set (Z, X) where Z is a vector of n centered variable of Y and X is n by m matrix of m predictors. $\tilde{Z}$ is the vector of estimated value where $\tilde{Z}_i$ is the estimated value of observation i defined as the weighted sum of all other $n-1$ responses.

$$\tilde{Z}_i = \sum_{j=1}^{n} w_{ij} Z_j \quad \forall\ i = 1, \ldots, n \tag{3.2}$$

The proposed influence measure can be expressed as:

$$\begin{aligned} \hat{I}_n &= cor(Z, \tilde{Z}) \\ &= \frac{\frac{1}{n-1} \Sigma_i Z_i \tilde{Z}_i}{S_Z S_{\tilde{Z}}} \\ &= \frac{\frac{1}{n-1} \Sigma_i \Sigma_j\ w_{ij} Z_i Z_j}{S_Z S_{\tilde{Z}}} \end{aligned} \tag{3.3}$$

where

$$S_Z^2 = \frac{1}{n-1}\Sigma_i Z_i^2 \tag{3.4}$$

$$S_{\tilde{Z}}^2 = \frac{1}{n-1}(\Sigma_i \tilde{Z_i}^2 - n\bar{\tilde{Z}}^2) \tag{3.5}$$

To ease the display, we define the following notations:

$$\Sigma_{(2)} = \sum_{i=1}^{n}\sum_{\substack{j=1 \\ i\neq j}}^{n} \tag{3.6}$$

$$\Sigma_{(3)} = \sum_{i=1}^{n}\sum_{\substack{k=1 \\ i\neq k\neq j}}^{n}\sum_{j=1}^{n} \tag{3.7}$$

$$\Sigma_{(4)} = \sum_{i=1}^{n}\sum_{\ell=1}^{n}\sum_{k=1}^{n}\sum_{\substack{j=1 \\ l\neq j}}^{n} \tag{3.8}$$

Theorem 3.2

Under the assumption stated in theorem 3.1, we have

$$(c_n^{-1} W_{S_{\tilde{Z}}^2})^{1/2} \hat{I}_n \xrightarrow{D} N(0,1) \tag{3.9}$$

where $W_{S_{\tilde{Z}}^2} = \frac{(n-1)}{n^2}\Sigma_{(2)} w_{ij}^2 - \frac{1}{n(n-1)}\Sigma_{(3)} w_{ik} w_{ij} - \frac{1}{n^2}\Sigma_{(3)} w_{ij} w_{kj} + \frac{1}{n^2(n-1)}\Sigma_{(4)} w_{i\ell} w_{kj}$

Proof:

To find the asymptotic distribution of $\hat{I}_n$, the two estimated value S_Z and $S_{\tilde{Z}}$ have to

be formulated.

$$
\begin{aligned}
E[S_Z^2] &= \frac{1}{n-1}\Sigma_i Z_i^2 = \sigma^2 \\
E[\tilde{Z}_i] &= E[\Sigma_j\ w_{ij}Z_j] = 0 \quad \forall i \in \{1,\ldots,n\} \\
E[\bar{\tilde{Z}}^2] &= E[(\frac{\Sigma_i \tilde{Z}_i}{n})^2] \\
&= E[(\frac{\Sigma_{(2)} w_{ij} Z_j}{n})^2] \\
&= \frac{1}{n^2}\ E[\Sigma_{(2)} w_{ij}^2 Z_j^2 + \Sigma_{(3)} w_{ij} w_{kj} Z_j^2 + \Sigma_{(4)} w_{i\ell} w_{kj} Z_\ell Z_j] \\
&= \frac{1}{n^2}[\Sigma_{(2)} w_{ij}^2 (\frac{n-1}{n}\sigma^2) + \Sigma_{(3)} w_{ij} w_{kj} (\frac{n-1}{n}\sigma^2) + \Sigma_{(4)} w_{i\ell} w_{kj} \frac{-\sigma^2}{n}] \\
&= \frac{(n-1)\sigma^2}{n^3}(\Sigma_{(2)} w_{ij}^2 + \Sigma_{(3)} w_{ij} w_{kj}) - \frac{\sigma^2}{n^3}(\Sigma_{(4)} w_{i\ell} w_{kj}) \qquad (3.10)
\end{aligned}
$$

$$
\begin{aligned}
E[S_{\tilde{Z}}^2] &= E[\frac{1}{n-1}(\Sigma_i \tilde{Z}_i^{\ 2} - n\bar{\tilde{Z}}^2)] \\
&= \frac{1}{n-1}(E[\Sigma_i (\Sigma_j w_{ij} Z_j)^2] - nE[\bar{\tilde{Z}}^2]) \\
&= \frac{1}{n-1}(E[\Sigma_{(2)} w_{ij}^2 Z_j^2 + \Sigma_{(3)} w_{ik} w_{ij} Z_k Z_j]) - nE[\bar{\tilde{Z}}^2]) \\
&= \frac{1}{n-1}(\Sigma_{(2)} w_{ij}^2 (\frac{n-1}{n}\sigma^2) + \Sigma_{(3)} w_{ik} w_{ij} (\frac{-\sigma^2}{n})) \\
&\quad -\frac{n}{n-1}(\frac{(n-1)\sigma^2}{n^3}(\Sigma_{(2)} w_{ij}^2 + \Sigma_{(3)} w_{ij} w_{kj}) - \frac{\sigma^2}{n^3}\Sigma_{(4)}(w_{i\ell} w_{kj})) \\
&= \frac{\sigma^2}{n}\Sigma_{(2)} w_{ij}^2 - \frac{\sigma^2}{n(n-1)}\Sigma_{(3)} w_{ik} w_{ij} - \frac{\sigma^2}{n^2}(\Sigma_{(2)} w_{ij}^2 + \Sigma_{(3)} w_{ij} w_{kj}) \\
&\quad +\frac{\sigma^2}{n^2(n-1)}\Sigma_{(4)} w_{i\ell} w_{kj} \\
&= \sigma^2(\frac{(n-1)}{n^2}\Sigma_{(2)} w_{ij}^2 - \frac{1}{n(n-1)}\Sigma_{(3)} w_{ik} w_{ij} - \frac{1}{n^2}\Sigma_{(3)} w_{ij} w_{kj} \\
&\quad +\frac{1}{n^2(n-1)}\Sigma_{(4)} w_{i\ell} w_{kj}) \\
&= \sigma^2 W_{S_{\tilde{Z}}^2} \\
& \qquad (3.11)
\end{aligned}
$$

Since $S_Z^2 = \frac{1}{n-1}\Sigma_{i=1}^n Z_i^2$ is a consistent estimator of σ^2, (3.11) implies $E[S_{\tilde{Z}}^2]$ can also be estimated by $W_{S_{\tilde{Z}}^2}(\frac{1}{n-1}\Sigma_{i=1}^n Z_i^2)$. The weight $W_{S_{\tilde{Z}}^2}$ is data dependent and different weighting mechanisms will lead to different values. A special case occurs when we do assume equal weights (i.e. $\frac{1}{n-1}$) on all the points beside observation i. The $W_{S_{\tilde{Z}}^2}$ is reduced to $\frac{1}{(n-1)}$.

By *Slutsky's theorem* and *theorem* 3.1, the asymptotic distribution of proposed measure $\hat{I}_n$ is approximated to a normal distribution as follows:

$$\begin{aligned}
(\frac{W_{S_{\tilde{Z}}^2}}{c_n})^{1/2}\hat{I}_n &= (\frac{W_{S_{\tilde{Z}}^2}}{c_n})^{1/2}\frac{\frac{1}{n-1}\Sigma_i\Sigma_j\ w_{ij}Z_iZ_j}{S_Z S_{\tilde{Z}}} \\
&= (\frac{W_{S_{\tilde{Z}}^2}}{c_n})^{1/2}\frac{\frac{1}{n-1}\Sigma_i\Sigma_j\ w_{ij}Z_iZ_j}{(\Sigma_i Z_i^2/(n-1))^{1/2}(W_{S_{\tilde{Z}}^2}\Sigma_i Z_i^2/(n-1))^{1/2}} \\
&= (c_n)^{-1/2}(\Sigma_i Z_i^2)^{-1}\rho_n \xrightarrow{D} N(0,1) \qquad (3.12)
\end{aligned}$$

Chapter 4

Backward Elimination Algorithm by I score

In this chapter, a heuristic backward variable selection algorithm based on the influence score is proposed. The algorithm aims to delete non-informative variables that will boost the influence score and return the variable set with the highest score during the elimination procedure. As the number of explanatory variables is large, due to the curse of dimensionality, we propose the backward eliminating algorithm based on random subset selection to detect important variable sets. The algorithm tends to keep the influential variables in the end; whereas, the noise terms are always identified and eliminated. In the following section, we will discuss the algorithm, the parameters related to the algorithm, and the algorithm's computational complexity. We also provide many different examples and applications based on different usages of the score.

4.1 Algorithm Based on Influence Score I

A large positive I score indicates the existence of informative variables included in current variable set and deleting noisy variables is likely to increase the score. To crystallize this property, we propose the following algorithm:

Algorithm 4.1: Backward elimination algorithm B times based on influence score

Step 1: *Randomly select a subset of d variables from total m dimensional variables. $\mathbf{X_d} = \{x_1, ..., x_d\}$ where x_i indicates the i^{th} variable of the selected subset. To avoid the curse of dimensionality, d is usually set as a moderate number such as between 5 and 10;*

Step 2.1: *To backward eliminate noisy variables within current d-dimensional variable set $\mathbf{X_d}$, compute the score $I(\mathbf{X_d})$ and $I(\mathbf{X_{d[-i]}})$ $\forall i = 1, ..., d$ where $I(\mathbf{X_{d[-i]}})$ represents the score computed without variable x_i. Delete j^{th} variable having maximum difference $I(\mathbf{X_{d[-j]}})$ - $I(\mathbf{X_d})$;*

Step 2.2: *If there is no variable remaining in the set, stop; otherwise repeat Step 2.1 with $d = d[-j]$;*

Step 2.3: *Return d_1 variables that attain the highest influence score as the returned variable set in the eliminating procedure;*

Step 3: *Repeat Step 1 - Step 2.3 B times;*

Step 4: *Do further analysis and applications (i.e. feature selection or classification) based on the returned variable sets with the highest B_1 ($B_1 << B$)* scores among the B repeat times.

Given a small set of total explanatory variables (i.e. m is not large), the algorithm can apply to the entire set and will return the most influential variable subset related to response. For large datasets, however, it is not practical to capture all high order interactions even when the repeat time B is set to an extremely high value. In some situations, to reduce the computational complexity, instead of backward eliminating the noisy features in d-dimensional subspace, we may use pairwise or triple-wise screening based on an evaluation of all lower order interaction effects.

Illustration of Backward Elimination Algorithm

The backward elimination algorithm is illustrated by the following simulated data set with response variable Y and independent variable set $\mathbf{X_7} = \{x_1, x_2, \ldots, x_7\}$ where all the x_i were generated independently from N(0,1) and x_1, x_2 are the only two influential variables. Consider the nonlinear relationship where Y is normally distributed with mean $(x_1 + x_2)^2$ and variance 1 for $N = 400$ observations. The two variables contribute both nonlinear and interactive effects to responses. The traditional correlation measure fails to identify these two influential variables since all absolute Pearson's correlations are smaller than 0.07. Among the variables, the maximum correlation is cor(Y, x_7) = 0.0643. Both of these two influential variables only show very weak signals, with cor(Y, x_1) = 0.0256 and cor(Y, x_2) = 0.0624, respectively.

By the algorithm, we first compute the joint influence score $I(\mathbf{X_7})$ with current variable set $\mathbf{X_7}$. To evaluate the influence of x_1, we remove variable x_1 and compute the new influence score $I(\mathbf{X_{7[-1]}})$ with the remaining variables. If this new score decreases substantially, it implies x_1 is important that similar responses Y cluster together in the space when it is included. On the other hand, increase of the new score indicates x_1 may contain noisy information in current variable set. We further compute new scores by removing one of the remaining variables. Every time, we eliminate the variable that boosts the score most. By repeating this procedure, we

continue discarding less informative variables until only one variable remains. Finally, the variable set that contains those variables leading to the highest influence score during the eliminating procedure is retained.

Table 4.1: History of the eliminating procedure for four cases with $k = 5$ $N = 400$

Initial set:{1,2,3,4,5,6,7}							
Influence before drop	0.6783	0.7968	0.8363	0.8503	0.8734	0.8883	0.4050
Dropped variable	6	7	4	5	3	2	1
Initial set:{1,3,4,5,6,7}							
Influence before drop	0.1419	0.2146	0.2549	0.2467	0.2561	0.4050	
Dropped variable	7	4	5	3	6	1	
Initial set:{2,3,4,5,6,7}							
Influence before drop	0.1429	0.2728	0.2952	0.2936	0.2928	0.3420	
Dropped variable	7	6	4	5	3	2	
Initial set:{3,4,5,6,7}							
Influence before drop	-0.0725	0.0374	0.0984	0.0504	0.0809		
Dropped variable	5	6	7	4	3		

In this simulated data set, we first centered the response value and then computed proposed influence measure although the score will not make any essential differences by centerizing Y. Table 4.1 presents the history of the change of influence score I and dropped variables step by step during the eliminating procedure. The first case shows the eliminating procedure if we include all seven variables in the initial set. Including all variables, we obtain $I(\mathbf{X_7}) = 0.6783$. In the next step, we compute all seven scores by eliminating each variable and removing the one that increases the score most. Therefore, in the first step, we dropped variable x_6 which led to an increase of the new score to 0.7968 with all the remaining variables $\{x_1, x_2, x_3, x_4, x_5, x_7\}$ to the next step. Continuing the algorithm, we observed that it attains the highest influence score when only x_1 and x_2 remained in the set. After one of the jointly influential variable x_2 is eliminated, the influence score dropped sharply, though it still maintain

in high value.

In the following, the algorithm is also applied to the different cases. We consider the cases that one of the two influential variables is not included in our initial set. We observed that the highest influence score attained when the only influential variable is kept in final step. In these cases, the algorithm will retain x_1 and x_2, respectively. Finally, we consider the case that only noninfluence variables are included in the initial set. The highest score is merely 0.0984 indicating the retained variable set does not contain much information. In addition, compared to these four cases, we observed that the initial score is the largest when both of the influential variables are selected in our initial set. When only one of the variable is present in the initial set, the initial score drops sharply to about 0.14, but is still larger than when none of influential variables are included. The initial score is the lowest one when the algorithm started by the set with all noninfluential variables. The negative starting value indicates the initial set has very poor predictive power and the score does not grow significantly by the algorithm. The highest score in the last case is smaller than any of the values when at least one of the influential variables is included. This simulated result also demonstrates the capability of proposed algorithm to detect influential variables if any of them are selected in the initial set.

4.2 Discussion on Repeat Time B

The backward elimination algorithm, depending on random sampling, is required to sample as many different combinations of the variables as possible. Assume there is an l-order interaction and it will pop out and be captured only when these l variables are selected simultaneously. In general, the repeat time B should be set large enough to capture the interaction effects, and it is related to the variable size of the data (m), the order of interaction (l) and number of variable selected(d) for each random sampling where $d << m$. Given a data set with m variables, to capture certain

l-order interaction by the algorithm with at least certain probability p, this implies the following inequality:

$$P(\text{capture } l\text{-order interaction}) = 1 - (1 - \frac{\binom{m-l}{d-l}}{\binom{m}{d}})^B > p \tag{4.1}$$

Therefore, we have

$$B > \frac{log(1-p)}{log(1 - \frac{\binom{m-l}{d-l}}{\binom{m}{d}})} \tag{4.2}$$

Table 4.2 shows the number of repeat time B required to catch potential l-order interactions. For example, when m=200, d=5 and p=0.75, we have to random sample at least 182,076 times to capture certain triple interactions. As both m and l get larger, the highly intensive computational burden may hinder the strategy. However, depending on different purposes, the algorithm is still worthwhile to apply. Furthermore, d value is inversely proportional to B and the effect will be shown in the next section.

Table 4.2: Number of Repeat Time B Needed

		p=0.5			p=0.75		
m	$d\backslash l$	2	3	4	2	3	4
200	5	1380	91038	>8.967*10^6	2759	182076	>1.793*10^7
	10	307	7587	213506	613	15173	427011
	15	132	2001	32847	263	4001	65694
500	5	8647	1435404	>3.566*10^8	17294	2870807	>7.133*10^8
	10	1922	119617	8492806	3843	239234	>1.698*10^7
	15	824	31547	1306586	1647	63094	2613171
1000	5	34623	>1.151*10^7	>5.741*10^9	69245	>2.303*10^7	>1.148*10^10
	10	7694	959818	>1.367*10^8	15388	1919636	>2.734*10^8
	15	3298	253139	>2.103*10^7	6595	506278	>4.206*10^7

4.3 Discussion on Number of Selected Variables d

In high dimensional dataset like gene expression microarray data, the number of observations, n, is usually small, only tens or hundreds. The number of variables m is usually very large ranging from several hundreds to more than ten thousand. The nearest-neighbor method suffers severely from the curse of dimensionality. Including all m variables together or using a large value of d will make the contained information diluted and slow the calculation of proposed algorithm. In addition, incorporating too much noise may lead to falsely eliminating true influential variables during the procedure. It is due to the reason that in a high-dimensional space, all points tend to be far away from each other. The nearest neighbors method based on the distance metric are not effective and meaningful.

To avoid the curse of dimensionality, we may only consider applying the procedure with a lower d-dimensional variable set, where d= 1, 2,...,$L \ll m$. In table 4.2, we observe large d eases the computational burden making the repeat time decrease remarkably. It is not always beneficial to increase d at will because the algorithm may fail to identify true interaction effects and influential variables in high dimension space.

Consider a simple triplet interaction example: $Y = X_1X_2X_3 + \epsilon$ with $N = 400$ where the variables and ϵ are $N(0,1)$. Including these three influential variables, different numbers of random Gaussian variables (i.e. d - 3 variables) are added to our candidate variable subsets. The simulations perform the proposed procedure with $k = 5$. We simulate the model 1,000 times to examine how different d affects the variable selection results.

In figure 4.1, the vertical axis is the proportion among the 1,000 simulations and horizontal axis is the number of total variables (d) included in our procedure. The solid line indicates the proportion of instances that the return set is exact the correct subset $\{X_1, X_2, X_3\}$ and dashed line shows the proportion of times that the final rough

set includes the exact set. It is obvious that both of the proportions deteriorate as the number of noisy variables increases. When d=5, the procedure is capable to identify the correct variable set accurately. In addition, as d is less than 20, the return set is still able to include the exact set and it has high possibility (>70 %) to precisely identify these three variables. If d is greater than 25, the algorithms starts to perform increasingly poorly and is gradually unable to include the exact set. When d starts at 100, although about 55 percent of the simulated results are able to include the exact set, less than 30 percent of them can identify the exact set.

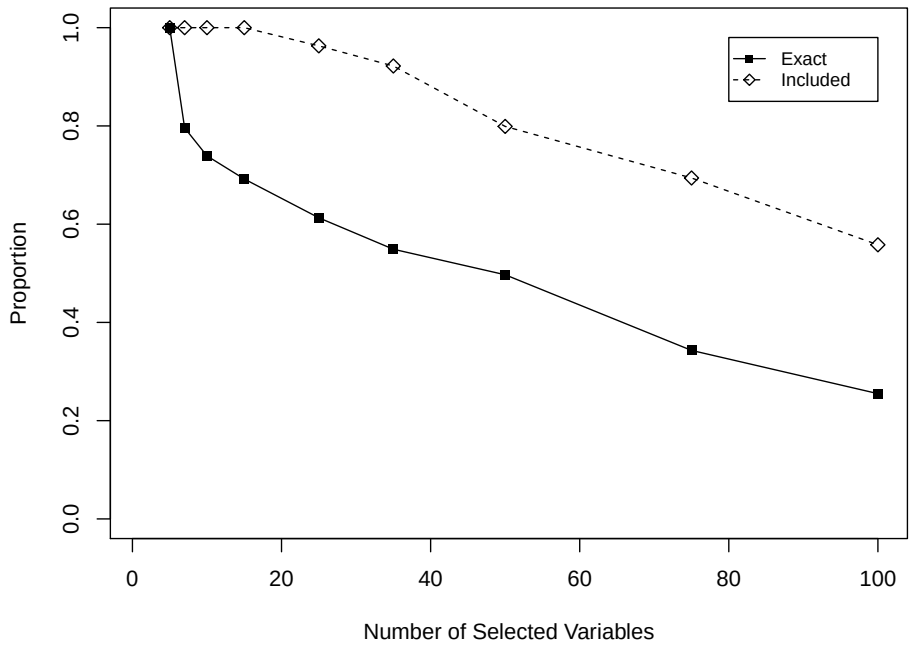

Figure 4.1: Proportion of influential variables left among 1,000 simulated return set

The mean and standard deviation of number of variables left in the 1,000 simulated return sets are shown in table 4.3. Both average and standard deviation grow as d increases; that implies more noisy variables are included in the return set. Compared to large d which may cause a deterioration of the proposed algorithm, smaller d leads

to relatively stable results.

Table 4.3: Summary of number of variables left in 1,000 simulated return set

	d								
	5	7	10	15	25	35	50	75	100
mean	3	3.229	3.331	3.458	3.976	4.963	8.059	16.541	27.432
sd	0	0.4824	0.6130	0.8324	2.1817	4.1209	8.3427	15.4703	20.7668

The simulations suggest that d should be set to a moderate number in order to improve the likelihood of identifying real influential variables. There is a trade-off between d and the number of total variables. For accurate identification of true influential variables, d should be set as a small value but the number of repeat times will increase significantly to capture the real effects. Furthermore, since the proposed algorithm has a higher chance to include true important variables, incorporating return frequency may be another option to supplement proposed procedure that is discussed in section 4.6.

4.4 Computational Complexity of Algorithm 4.1

As discussed in the previous sections, many factors will affect the computational complexity, including number of observation (n), number of selected variables (d) and repeat time B. The algorithm depends on the random sampling and there is a high chance that the random subsets may not contain any important information. In general, the repeat time should be set large enough and only the top B_1 highest score subsets should be chosen for further applications. The computational complexity required to find the top B_1 informative variable subsets involves two main parts: B times backward elimination algorithm and sorting the B variable subsets by the influence scores.

B times backward elimination algorithm

In each random subset, the complexity of backward elimination algorithm includes nearest neighbor searching, estimating the target, and computing the influence score with a decreasing number of dimensions. For n observations, not taking all m variables into account, we only randomly select d variables ($d << m$). The computational complexity of the influence score given d dimensional space includes:

(a) The distance calculation of each observation to all remaining (n-1) points is $O((n-1)d)$. For all n observations, the computational complexity is $O(n^2d)$ (i.e. $O(\frac{n(n-1)}{2}d)$)

(b) Sorting by the distance to find the nearest neighbors of every observation can be found in $O(nlogn)$ time (i.e. $O(kn)$ if $k << logn$) and it takes $O(n^2logn)$ for all n observations.

(c) Estimating and computing the influence score with all n observations leads to a the complexity that is $O(n)$

Combining all the computations, the total complexity is $O(n^2d + n^2logn + n)$ where the search of k nearest neighbors of all observations can be found by the sum of computations in (a) and (b) with $O(n^2(d+logn))$ time. The time complexity in (c) is negligible compared to nearest neighbors computation. Hence, the approximated time for computing the influence score with d variables is $O(n^2(d+logn))$ (or $O(n^2(d+k))$ if $k << logn$)

The time complexity in each random subset is computed by the following ways. In the first step, the influence score is computed only once with all d variables. To find the one irrelevant variable at a time, the procedure is repeated with the next step computing the score with $d-1$ variables d times, so that each variable removed once. Therefore, the computational complexity with $d-1$ variables is $O(d(n^2(d-1)+n^2logn))$. Continuing to eliminate each less informative variable until the last stage with one variable remains, the total complexity for one backward elimination

algorithm is $O((d+d(d-1)+(d-1)(d-2)+\ldots+2)n^2+(1+d+(d-1)+\ldots+2)n^2logn)$, which is bounded by $O(d^3n^2+d^2n^2logn) = O(d^2n^2(d+logn))$ (i.e. $O(d^2n^2(d+k))$ if $k << logn$). In a high dimensional data set such as microarray, n is usually not very large with $n << m$. Setting d to be a smaller number like 5 to 10 or $log(n)$ will ease the computational time and also avoid curse of dimensions. If we set $d = logn$, the computation time for the algorithm is bounded by $O(n^2log^3n)$.

For B times backward elimination algorithm, the complexity for the first part is $O(Bd^2n^2(d+logn))$. In order to explore highly informative subsets, the repeat time B should be set to a large value in order to make the random sampling cover as many combinations of variables as possible.

Finding the top B_1 informative variable subsets

In general, $B_1 << B$, to find the top B_1 variable subsets is just sorting for the B variable subsets and it takes $O(BlogB)$.

Combining these two terms together, the overall computational time is $O(Bd^2n^2(d+logn)) + O(BlogB) = O(B(d^2n^2(d+logn)+logB))$. In generally, $logB << d^2n^2(d+logn)$, the complexity for finding the top B_1 building blocks is bounded by $O(Bd^2n^2(d+logn))$.

4.5 Backward Elimination with Different k

Consider there to be two classes having response variable Y which behave differently. Given there are 400 observations and the number of classes is selected from a binomial distribution with probability 0.6 to be first class. Suppose that the data has independent variable set $\mathbf{X} = \{X_1, X_2, \ldots, X_{10}\}$ where all X_i are generated independently from N(0,1). The response value are generated by $Y_i = X_{i1}X_{i2} + \epsilon_i, i = 1, \ldots, n_1$ and $Y_i = (1-X_{i3})(1-X_{i4}) + \epsilon_i, i = n_1+1, \ldots, n$ for class 1 and class 2 respectively where ϵ_i is N(0,1). There are four influential variables and each class is affected by different pairs of independent variables.

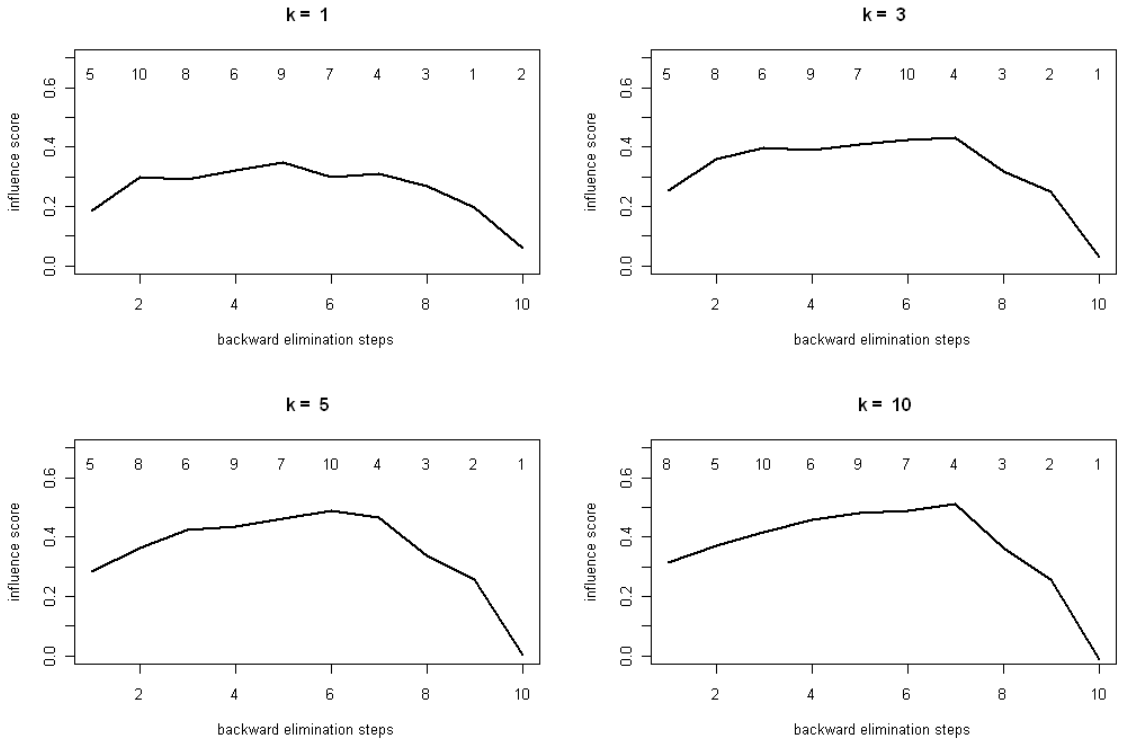

Figure 4.2: Backward elimination with various parameter k

Applying backward elimination algorithm to the simulated data set, figure 4.2 shows the backward elimination result with different numbers of nearest neighbors. The number in the upper side of each box indicates the eliminated variables step by step. We observe that all the irrelevant variables $\{X_5, \ldots, X_{10}\}$ are eliminated in the first few steps across all the different k. When $k = 1$, the initial influence score with all $\mathbf{X}$ is 0.1859. After removing X_5, the influence score increases to 0.2993 and so on. However, the measure does not increase a lot but fluctuates in this case since using 1 nearest neighbor may include more noise and it may not be reliable. The procedure with k=1 reaches the peak of 0.3492 after eliminating $\{X_5, X_{10}, X_8, X_6\}$. The score starts to decrease and drops sharply once informative variables are eliminated in step 7.

The influence score is generally become larger as k increases. For example, the ini-

tial and peak scores are higher with larger k compared to smaller one. In addition, the score drops sharply with larger k than when $k = 1$ once any informative variables are discarded. In general, the influence score curves have similar pattern across different k and the four influential variables are always kept after step 6 ($i.e.$ after eliminating all non-informative variables.) If we select the remaining variables as the score attains the peak with different value of k, all the influential variables are returned by applying our algorithm. However, the exact influential variable set is obtained when k= 3 or 10. For k=1 and 5, the return set will include additional variables $\{X_9, X_7\}$ and $\{X_{10}\}$ correspondingly. We also notice the influence scores decrease to their lowest value when only 1 variable is contained in the end, though it is a true influential variable, the joint effect disappears making the individual variable non-informative. This is caused by signal from interactive but not a marginal effect. The influential variable set is impossible to be identified if we apply marginal method without prior knowledge of the functional form. For example, by the Pearson's correlation, the signal is very weak X_1 (-0.007), X_2 (0.021).

4.6 Return Frequency in Screening

With moderate sample size n and number of variables m, the return frequency is another choice to screen important variables. As shown in section 4.3, the influential variables are high likely to be included in the returned set. If some variables have strong marginal effects or may only appear when considering the joint effects, the variables will consistently be returned with higher scores. Therefore, we develop another strategy by computing the top list pair or triplet and select variables with higher ranks of return frequency. It is not practical to simply report the top score pair or triplet as influential variables since a strong marginal effect variable is likely to carry some noninfluential variables with it. Return frequency will be a conservative choice for the purpose of avoiding the selection of false signals that would appear with

strong marginal effect.

Consider we randomly assign c = 6 different clusters and the number of each cluster is distributed by multinomial distribution with probability (0.3, 0.2, 0.15, 0.15, 0.1, 0.1). The response values are normally distributed with center $\bar{y_{c_i}}$ from uniform(0, 5) and $\sigma^2_{c_i}$ from uniform(0.3, 1.5). As for the influential variables, let P_1 be a randomly selected orthonomal matrix of dimension m by m matrix (*i.e.* $m = 10$) among S=500 variables. The center C_i for $1 \leq i \leq c$ with m dimensional space have distribution

$$C_i = P_1 D_1^{1/2} W_i$$

where D_1 is a diagonal matrix with the elements independently distributed as exponential with mean 1 and all W_i are independent $N(0, 1)$. Conditional on P_1 and D_1, the center C_i comes from multivariate normal distribution $\mathbb{N}(0, P_1 D_1 P_1')$. Given the i^{th} cluster c_i, the influential variables X_{c_i} are generated as

$$X_{c_i} = C_i + P_2 D_2^{1/2} W_{c_i}$$

where D_2 has the same distribution as D_1 and P_2 is another randomly orthonormal matrix of dimension m by m. W_{c_i} are also independent $N(0, 1)$. In this setting, a few of influential variables may have effect due to interactions. In addition, the remaining 490 variables are random noise from $N(0, 1)$.

It is easy to compute exhaustively such as marginal and pairwise scores. For the higher dimensions interaction like triplet are also possible to calculate. Here, we compare the selection results for different methods. First, I_1 is computed based on marginal influential scores ($d = 1$). I_2, pairwise score, is computed for total $S(S\text{-}1)/2$ pairs. To evaluate the rank of individual variables based on pairwise score, we count the return frequency (*i.e.* rI_{2f}) of an individual variable appearing in the top n_r scores where n_r is an essential portion of the number of total pairs. In this example, $n_r = 5000$ is used.

The ranks of influential variables with different sample size (n=200, 400) are listed in Table 4.4 with marginal correlation and $rI1$, rI_{2f}.

Table 4.4: Rank of influential variables

Variable	1	2	3	4	5	6	7	8	9	10
n=200										
$r\|cor(y,x_i)\|$	1	5	2	340	138	4	11	3	104	17
k=1										
rI_1	1	19	23	5	204	122	318	10	55	352
rI_{2f}	1	7	3	2	11	5	101.5	4	6	19
k=3										
rI_1	1	25	3	2	50	177	369	11	114	372
rI_{2f}	1	7	2	3	8	5	34	4	6	10
k=5										
rI_1	1	11	3	2	24	104	295	10	75	265
rI_{2f}	2	7	2	2	8	5	39	4	6	11.5
k=10										
rI_1	1	4	3	2	10	20	162	5	44	90
rI_{2f}	2	7	2	2	12	5	35	4	6	8
n=400										
$r\|cor(y,x_i)\|$	1	5	3	485	59	4	6	2	83	29
k=1										
rI_1	1	48	2	123	218	115	130	4	3	10
rI_{2f}	1.5	5	1.5	6	8	9.5	7	3	4	9.5
k=3										
rI_1	1	9	2	17	98	86	90	4	3	16
rI_{2f}	1.5	5	1.5	7	9	8	6	3	4	10
k=5										
rI_1	1	5	2	10	49	61	42	3	4	19
rI_{2f}	1.5	5	1.5	7	9	8	6	3	4	10
k=10										
rI_1	1	5	2	8	26	18	13	3	4	15
rI_{2f}	2	5	2	6	9	8	7	2	4	10

As $n = 200$, the marginal correlation methods identify some of the strong marginal effects but fail to identify other effects such as X_4, X_5, X_9, X_{10}, where the ranks are all much higher than 10. These might be due to non linear or interactive relationships.

On the contrary, with one dimensional influence measure rI_1, X_4 is ranked as 5^{th} indicating a strong non-linear effect which will only be identified by proposed measure. In general, the marginal and high degree (i.e. polynomial) effects can be identified by rI_1 with moderate k. However, some variables such as X_5 and X_9 show strong signals only when we consider rI_{2f}. The effects may come from interactions. With a small number of k, proposed influence score does not perform well compared to the marginal method. The result improves as k becomes larger in both rI_1 and rI_{2f}. It is also obvious that rI_{2f} is overall better than rI_1 with different k although one or two variables may fail to be selected if the threshold of rank is set stringently. For example, in k=10, selecting the 10 highest rank variables will lead to missing two influential variables X_5, X_7.

As sample size n grows to 400, the marginal correlation method shows very consistent results as that with $n = 200$. rI_1 also shows some obvious improvement as k gets large. For rI_{2f} with large sample size, the signals of influential variables are strong enough to be captured with different k since all the ranks of these variables are within the top 10. With larger sample size, rI_{2f} computed by smaller k is sufficient to find influential variables from various types of effects, but rI_1 perform well only with larger k.

Furthermore, table 4.5 lists the ranks of a few non-influential variables $X_{11}, \ldots, X_{20}$. We observed all the variables do not exhibit strong signal in any of the methods since almost all the ranks are very high. Generally, with larger sample size and moderate k, return frequency computed from high order interactive scores have strong power to discriminate the true influential variables from noisy variables.

Table 4.5: Rank of non-influential variables

Variable	11	12	13	14	15	16	17	18	19	20
n=200										
$r\|cor(y,x_i)\|$	240	497	347	372	431	449	206	176	378	454
k=1										
rI_1	351	94	183	136	128	469	453	74	174	284
rI_{2f}	400.5	11	94.5	158.5	133.5	80	400.5	19.5	346	68.5
k=3										
rI_1	349	37	169	69	76	253	454	150	258	331
rI_{2f}	441.5	12	82.5	239	121.5	97	239	27	121.5	46.5
k=5										
rI_1	485	189	241	266	207	249	312	104	326	149
rI_{2f}	306	42	226	63	84	245	397	288	348	394
k=10										
rI_1	466	92	210	198	246	115	192	38	302	122
rI_{2f}	142	94	404	123	154	221	411	417	362	432
n=400										
$r\|cor(y,x_i)\|$	286	270	86	93	182	200	137	240	173	29
k=1										
rI_1	9	451	366	362	206	386	238	343	40	299
rI_{2f}	277	234	234	330.5	43.5	234	114.5	454	277	234
k=3										
rI_1	145	412	385	160	73	342	241	320	56	261
rI_{2f}	329.5	380	191.5	234.5	22.5	191.5	88	431	431	234.5
k=5										
rI_1	237	405	444	94	26	335	209	352	110	177
rI_{2f}	306	306	159.5	361.5	22	250	54	418.5	465	199.5
k=10										
rI_1	192	433	417	67	24	284	90	373	288	81
rI_{2f}	210.5	210.5	157.5	275.5	33.5	121	79.5	346	346	413.5

Chapter 5

Application to Classification Problems

In previous chapters, we show that the influence score I is a measure of association between a subset of variables and continuous responses. It can be applied to classification problems with dichotomous responses and continuous predictors like microarrays. It is known that there are quite a number of influential genes in gene expression datasets and many of them contribute somewhat to the diseases in different ways. In addition, many studies show that epistasis occurs when a certain gene is modified or regulated by one or more modifier genes (Phillips, 2007). The phenomenon may imply potential functional interaction of certain gene sets in complex diseases. In this chapter, we discuss and apply the proposed influence score I with inverse rank weight function to screen informative building blocks (i.e. influential variable sets) in classification problems and study a new classification procedure in three gene expression data sets.

5.1 Illustrations of Influence Score in Two Class Problems

The proposed influence score can be applied to screen potential important variables with both marginal and interactional effects. The main advantage of using it is similar to that of filtering method in classification, they are benefit from their simplicity and computational efficiency. The screening method, unlike other methods using searching algorithms in possible subsets of variable space, needs less computational time. Another advantage is the capability to catch those variables with nonlinear relationship with the output labels, especially specific local structure. When we use other linear methods, such as Pearson's correlation, to screen the nonlinear variables that will result in a lower score, we then treat them as noisy variables leading to a loss of some important signals. With the new influence measure, the nonlinear and important effects will be captured. Furthermore, the proposed score is able to detect high order joint effects. In the following, we illustrate proposed scores in the context of two examples.

5.1.1 One Dimension - Marginal Effect

Similar to traditional filtering methods, the influence measure also provides an efficient way to do screening as the number of variables is large. After calculating the scores of the variables, the ranks of the variables are obtained by sorting these scores. Those of top rank variables can be used as informative predictors to build the classifiers. Most of the existing filtering methods are independent of the choice of the classifier and are focusing on global patterns such as a linear separable condition. In reality, it is impossible to find a clear cut example of a two class problem. Therefore, the proposed influence measure takes advantage of neighborhood information, which may help identify variables with both global and local patterns.

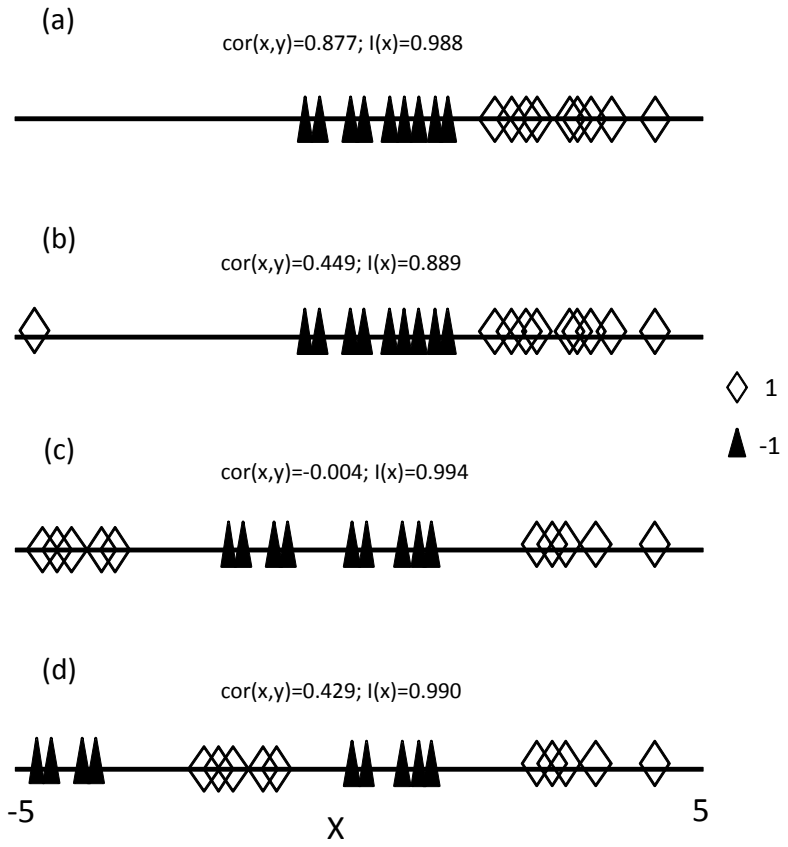

Figure 5.1: Two class problem in different one dimensional examples with Pearson's correlation and influence scores by k=5

Consider a few one dimensional distributions of two-class data set in figure 5.1 where the two difference classes with predictor X distributed in the range (-5, 5). The correlation score and proposed $I(\mathrm{X})$ score with $k = 5$ are computed. In (a), the data can be clearly separated by any points between the two classes. Both the Pearson's correlation (0.877) and proposed influence score (0.988) are high enough to indicate the importance of X. However, if there is an additional outlier such as (b), the pattern is still very similar to (a). The correlation score drops a lot from 0.877 to 0.449; and that may lead to neglecting the variable although the pattern is still very

clear. As for $I(\mathrm{X})$, it drops a little to 0.889 and the outlier only has a minor effect such that we can still catch the signal. In (c), there are three clusters of the two classes. The right and left ends belong to the same class. The Pearson's correlation is close to 0 but I maintains a large value to indicate the significance of the predictor. Actually, the marginal nonlinear effect in this example is very clear but we may ignore it by traditional screening methods. For (d), four clusters are distributed alternatively with the two classes. The Pearson's correlation score is moderate, and may fail to be treated as informative but the high score of $I(\mathrm{X})$ implies specific local patterns may exist. From these different situations in figure 5.1, we observe the proposed I score is able to capture important marginal effects of both global and local patterns in two class problem. In addition, the existence of an outlier effect will be alleviated by the proposed I score.

Once we identified the important predictor, in (a) and (b) of figure 6, almost all of the existing classifiers are capable of strong performance. However, in(c) and (d), only the classifiers taking local patterns into consideration such as tree, k nearest neighbors can do well. Therefore, to take advantage of the proposed I score after screening out the informative variables, those classifiers adopting both global and local structures are suggested for further applications.

5.1.2 Two Dimensions - Marginal and Joint Effects

In this section, we illustrate classification problem with two input variables. The first one involves only one variable having marginal effects and the second is the XOR problem with strong joint effects. The simulated predictors are from different bivariate normal distributions with mean values set to be (0, 0) and (0, 3) for class 1 and class 2, respectively. In addition, the standard deviations of both variables are 1. The number of observations in each class is 200.

Figure 5.2 shows the distributions of these two classes. The absolute Pearson's

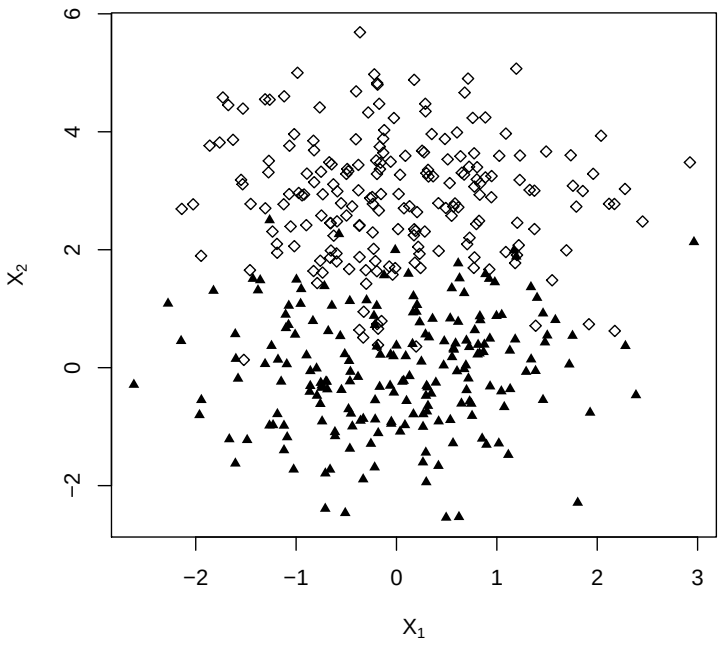

Figure 5.2: Scatterplot of two dimensional problem with one variable associating with class labels.

correlation of each individual variables is $|c(X_1)| = 0.0049$ and $|c(X_2)| = 0.8169$. For our influential score with $k = 5$, $I(X_1) = 0.0027$, $I(X_2) = 0.8586$ and $I(X_1, X_2) = 0.8116$. From the figure, X_1 is pure noise if the class label is projected to X_1 that leads to both $c(X_1)$ and $I(X_1)$ are very low. Since X_2 is linearly correlated with the target labels, the two scores corresponding to $c(X_2)$ and $I(X_2)$ are relative high. The joint score $I(X_1, X_2)$ is still very high, however, due to including the noisy variable X_1, which makes the score a little lower than that when only considering X_2.

For non-linear relationships as in figure 5.3, the XOR classification problem is generated. There is no prediction power when considering each variable individually, but it will be boosted when both of them are included. We can find the scores with only one individual variable are all very low $c(X_1) = 0.0092$, $c(X_2) = 0.0179$, $I(X_1)$

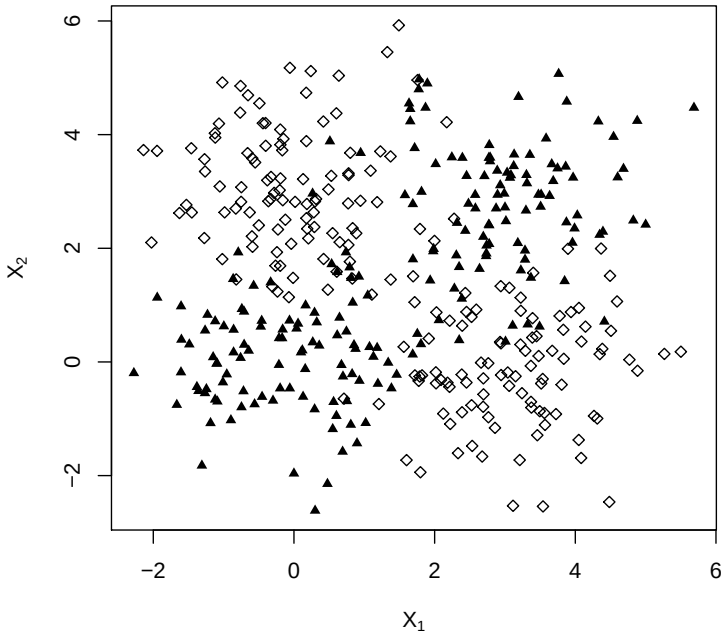

Figure 5.3: Scatterplot of two dimensional problem with joint effects: XOR problem.

$= 0.1150$ and $I(X_2) = 0.0051$. The influence measure is flexible to compute joint influential score $I(X_1, X_2) = 0.7643$. It is relatively high compared to the case when only a single variable is included.

In summary, both correlation and influential score can distinguish clearly between a noisy variable and the one with linear association with target values. The proposed score is able to detect strong nonlinear effects and especially boosts the ability to identify joint effects of a variable subset in two dimensions and more. In reality, not only pairwise but higher order interactions play an important role in classification problems, especially gene expression microarray. The backward elimination algorithm will help detect higher order interaction sets called building blocks. We can further construct a strong classifier by aggregating those highly informative building blocks, in an effort to do prediction.

5.2 Microarray Data Set

In complex diseases, as in microarray data, not only marginal and pairwise interaction effects but various combinations of genes may form an influential set. Many studies (Li, 2009; Oti, 2007) show pathway and genes are functionally related, contributing to the causes of complex diseases. These studies suggest higher order interactions may benefit prediction.

In this study, three gene expression datasets are analyzed via our proposed procedure. The first dataset is a breast cancer data first analyzed in Van 't Veer et al. (2002) and re-analyzed in Tibshirani and Efron (2002). There are 4918 genes included in this data set. The training portion of the data set contains 78 patients, 34 of which are patients who had developed metastases within 5 years (relapse). The remaining 44 samples are from patients who remained healthy from the disease after their initial diagnosis. In addition, the testing set contains 12 relapse and 7 non-relapse samples. Overall, 4918 gene expression profiles for 51 good prognosis breast cancer samples (non-relapse) and 46 poor prognosis breast cancer samples (relapse) are included. The second data set involves prostate cancer analyzed in Singh et al., (2002). The data set consists 12,533 gene expression profiles for 52 prostate tumors and 50 non-tumor prostate samples. The third dataset is a collection of gene expression measurements from colon cancer reported by Alon et al., (1999). This dataset is relatively unbalanced since it consists of 42 tumor tissues and 22 normal colon tissues, and the final assignments of the disease status were made by pathological examination. Originally, 6000 genes are measured by high density oligonucleotide arrays and 2000 genes across 62 samples were selected based on the confidence in the measured expression level.

5.3 Classification Procedure

Figure 5.4 shows our proposed classification procedure. It includes many parts. First, we perform an interaction-based screening by pairwise scores to select variables and, second, we apply the backward elimination algorithm to find informative sets of variables (building blocks) and filter these sets to find independent building blocks. Finally, we aggregate them to form a stronger classifier by boosting. We illustrate this procedure in breast cancer data with a 78 patient training set to find the building blocks, and tune the weights in boosting algorithm to aggregate them by different classifiers and predict the 19 patient testing set. Furthermore, we use 10-fold cross-validation to evaluate the performances of proposed procedure in three microarrays.

5.3.1 Pairwise-Based Screening

In section 4.1, we propose a backward elimination algorithm to screen important building blocks (i.e. subset of variables). The purpose is to find the most informative sets to assure the capability to accurately predict the outcome. In section 4.2, to capture complicated patterns in complex data set, we show the number of repeat time B will increase if too many candidate variables are included. In high dimensional data set, due to the large number of variables in gene expression datasets, directly applying the algorithm with all genes without preliminary screening is not efficient. Important information may be missed due to its chance to be selected and eventually diluted by including many noise variables. In the following, the 78 training set subjects with 4918 genes of breast cancer training set are used to illustrate the procedure of screening building blocks and constructing a weighted classifier by boosting algorithm.

We carried out the variable screening based on the return frequency of pairwise influence scores discussed in section 4.6 in order to filter informative subset of candidate variables. The higher return frequency of a variable, the stronger implication that it

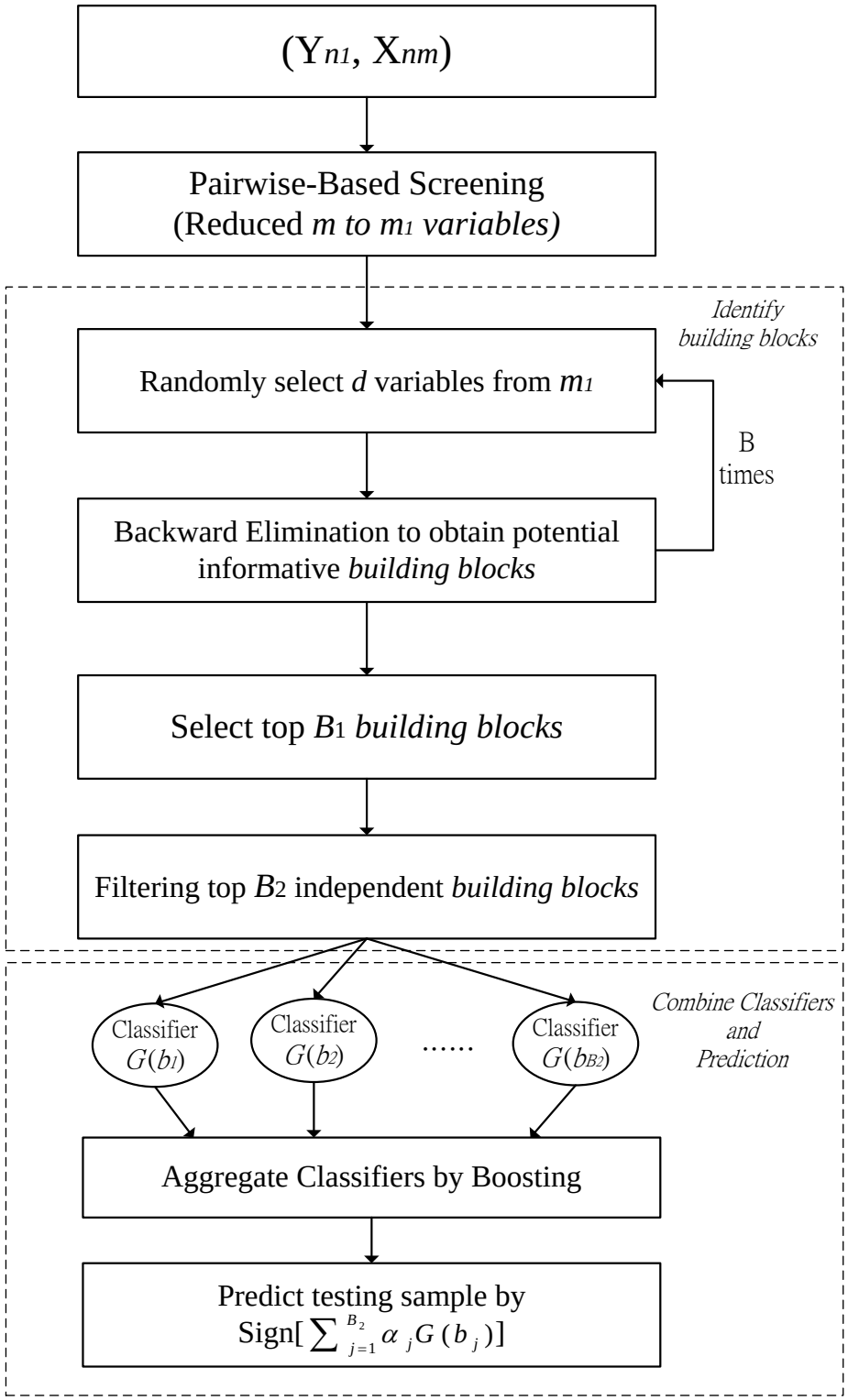

Figure 5.4: Procedure of classification: identifying independent building blocks and aggregating classifiers

has the potential to be an important variable and to form an informative building block. Given 4918 genes in the data set, there are more than 12 million pairs. We choose the top 1% pairwise scores and count the return frequency of the 4918 genes to pre-select the genes with highest return frequency and reduce the number of variables to be an arbitrary value 300 for different screening parameter k=(1,3,5) during the pairwise screening step.

Compared to the regular absolute Pearson's correlation measure, we observe more than half among the top 300 variables are different from the top 300 variables chosen by return frequency computed by I_r with varied k. We also observed a few genes with weak marginal effects but having very high return frequency, which implies the existence of nonlinear or joint effects. For example, gene 4836 has absolute Pearson's correlation 0.1112 that ranked 1159 among 4918 genes. However, its return frequency is extremely high for $k = (1, 3, 5)$. That (202, 343, 373) are all ranked within top 25 among the returned variables. The reason that the Pearson's correlation is small might be due to an outlier (i.e. training sample 54) of gene 4836 that is 6 standard deviations away from the mean. It affects the computation of correlation based on the assumption of linear relationship. Our proposed influence measure can ease such effects since the assumption is relaxed.

5.3.2 Identifying Building Blocks

By setting the repeat time to be a large value (B=5,000,000) and a moderate value of d=6 (i.e. $\lfloor log_2 n \rfloor$), the backward elimination algorithm is applied with varied $k = (1, 3, 5)$. Setting d as a moderate value has many advantages. First, to avoid the curse of dimensionality. Second, to facilitate the computation time. Third, we avoid the false elimination of influential variables by including too many variables to the algorithm. The top 20 unique influence scores identified by backward elimination algorithm and subsets of variables by $k = 5$ are listed in table 5.1. Some clusters of

Table 5.1: Top 20 scores of marginal, pairwise and informative building blocks in breast cancer data with $k = 5$

	Marginal		Pairwise		Backward Elimination Algorithm	
Rank	Genes	Scores	Genes	Scores	Genes	Scores
1	450	0.5155	2819,4055	0.6945	1904,2737,3481,4055,4705	0.8210
2	4055	0.5110	45,4096	0.6815	549,1951,4138,4836,4913	0.8103
3	1419	0.4642	1148,2294	0.6786	11,4040,4096,4705	0.8072
4	4570	0.4477	2729,4055	0.6785	323,2335,2751,2826	0.7986
5	4868	0.4442	2895,4096	0.6749	11,3423,3733,4040	0.7959
6	798	0.4418	1953,4096	0.6702	400,2294,3355,3807	0.7907
7	2294	0.4405	3154,4055	0.6665	934,1859,4096,4533	0.7869
8	3495	0.4384	4055,4101	0.6619	489,1450,3004,4305	0.7863
9	3897	0.4334	3758,4405	0.6595	298,488,2006,4096	0.7799
10	1241	0.4313	618,3398	0.6579	787,1935,4096,4533	0.7790
11	4405	0.4287	323,2751	0.6562	323,798,2751,3722	0.7789
12	45	0.4243	1818,3733	0.6537	323,1175,1744,2751	0.7768
13	3358	0.4208	437,1969	0.6537	400,2183,3657,4096,4204	0.7766
14	1175	0.4208	400,3807	0.6534	400,3484,3807	0.7762
15	1951	0.4200	4055,4076	0.6528	323,475,2888,4740	0.7756
16	4308	0.4182	2888,4405	0.6524	323,2025,2230,4378,4916	0.7752
17	3649	0.4172	3381,4055	0.6516	1374,3484,3499,4096	0.7729
18	4014	0.4160	849,1404	0.6513	323,2025,3423,3733	0.7707
19	298	0.4131	4096,4570	0.6512	400,989,2470,3807	0.7706
20	4374	0.4106	3040,4096	0.6507	2737,3481,4055,4226	0.7698

Pairwise: Top 20 scores of all pairwise combinations (12,090,903 pairs)
Backward Elimination Algorithm: Top 20 scores with B=5,000,000 and d=6

variables among the top 20 are returned many times, but we only list the unique one. The table also includes the top 20 marginal and pairwise influence scores. We can easily observe that the top scores increase greatly when taking higher order interactions into consideration.

Among the 4918 variables, the two strongest marginal influence scores are 0.5155 and 0.5110 for gene 450 and gene 4055. We also observed that many high pairwise score pairs are related to gene 4055. In addition, combining two strong marginal effects does not guarantee higher pairwise effects since the highest pairwise score is

a confluence of one strong and one moderate marginal effects. For example variable 2819 has marginal score only 0.1969 but the joint score with gene 4055 increases to 0.6945. In addition, the proposed algorithm shows that many strong joint effects of higher order interaction exist in the top informative set since all the top 20 scores are all greater than the highest pairwise influence score 0.6954.

We also observe some combinations of genes appear many times by proposed algorithm. The rank 11^{th} and 14^{th} building blocks appear 9 and 23 times, respectively. This also indicates the strength of the joint effects of such gene sets. They are always returned once the combinations of such gene sets are included in random sampling.

The informative building blocks usually have size between 3 to 5 genes. Some of them consist of genes with only moderate marginal effects but showing strong joint effects. For instance, the 3^{rd} subset has joint score 0.8072, but all these four genes (11, 4040, 4096, 4705) do not have very strong marginal scores (0.2898, 0.1591, 0.3668, 0.0541). If we only consider gene selection with marginal methods, due to the weak marginal signal, many variables would be eliminated and we would not have been able to find such a strong building block. The joint effects may play an important role since gene 4705 survives in the candidate variable set by pairwise based screening.

The ideal aggregated classifier is to have the basic units of building blocks uncorrelated. We found there are many overlapped variables among the top 20 building blocks like genes 323, 400, 4055, 4096. We should not directly aggregate all of the top informative building blocks to make our final classifier since they have common variables leading to correlated issue. Instead, we only keep one of those return building blocks containing common variables. This can be completed by removing those having variables in common with higher scored building blocks. If we set the threshold at 0.7, there are tens of thousands of building blocks that have scores greater than 0.7. After the filtering procedure with common variables, only a few dozen building blocks remain.

5.3.3 Classification Algorithm

Methods of classifier Aggregation

There are many methods developed to aggregate classifiers to produce a final prediction. The most simple one is majority vote with equal weight which is based on the law of large numbers that follows if the classifiers are independent. Aggregating the classifiers will improve prediction accuracy if their capability to pinpoint the right decision is better than random guessing. Given there are M building blocks based on the same classifier $G(.)$ in two class problem, each of them forms a classifier $\hat{G}(b_1)$, $\hat{G}(b_2)$, ... , $\hat{G}(b_M)$, the prediction rule can be expressed as:

$$\hat{G}^* \quad = \quad sign\left(\sum_{m=1}^{M} \hat{G}(b_m)\right) \tag{5.1}$$

If each independent classifier has the same probability $p \in [0, 1]$ to make a correct decision, the probability to do the right prediction called *Condorcet's jury theorem* is

$$Pr\{majority\ make\ correct\ decision\} \quad = \quad \Sigma_{\lfloor \frac{M}{2} \rfloor + 1}^{M} \binom{M}{i} p^i (1-p)^{M-i} \tag{5.2}$$

In addition to equal weight, stacking is another model averaging method that searches the best weights $w = (w_1, w_2, \ldots, w_M)$ among the classifiers by solving the quadratic programming problem:

$$\begin{aligned} &\hat{w} = argmin_w \ \Sigma_{i=1}^{N}[y_i - \Sigma_{m=1}^{M} w_m \hat{G}^{-i}(b_m)]^2 \\ &s.t. \quad \sum_{m}^{M} w_m = 1 \\ &\qquad w_m \geq 0 \ m = 1, 2, \ldots, M \end{aligned} \tag{5.3}$$

where $\hat{G}^{-i}(b_m)$ m = 1,2...M are estimated by q-fold cross-validation and the final prediction function is $\Sigma_{m=1}^{M} \hat{w}_m \hat{G}(b_m)$. Therefore, we avoid giving unfairly high weight to specific classifier with higher complexity. However, the key drawback of this aggregating algorithm is the singularity when we search the best weights by solving

quadratic programming. This may be due to either perfect classification of a few strong classifiers from a cross-validation estimate, or identical estimates of misclassification rate from different classifiers. In such situations, the weights may concentrate on only a few classifiers that make the "optimal weights" subject to high variability and may mean they might no longer be optimal.

Boosting (Schpaprie 1990, Freund and Schapire, 1997), one of the most powerful learning methods, is adopted to aggregate the important building blocks. One advantage is that many studies observed empirically that boosting does not overfit the data (Leo Breiman, 1998; Drucker et al, 1996). Boosting builds an additive model that sequentially adds one classifier to reweighted versions of the training data, and takes the weighted majority vote of the selected sequence of classifiers:

$$G(b) \quad = \quad \sum_{m=1}^{M} \alpha_m G(b_m) \tag{5.4}$$

where the impact parameters of each classifier $\alpha_1, \alpha_2, \ldots, \alpha_M$ are constants tuned by the boosting algorithm. The higher the alpha is, the more informative the building block in the sequence. Ideally, the parameters $\{\alpha_i, \forall \text{ i} = 1, 2, \ldots, M \}$ are generated by minimizing an exponential loss function:

$$min \sum_{j=1}^{N} L(y_j, G(b)) = min \; E(e^{-yG(b)}) \qquad y \in \{1, -1\} \tag{5.5}$$

where

$$G(b) = \frac{1}{2} \; log \; \frac{P(y = 1|b)}{P(y = -1|b)}$$

The solution is approximated by iteratively adding a single building block one at a time to the aggregated model without adjusting the parameters of those having already been included. That is, when adding new building blocks k+1, we minimize

$$\sum_{j=1}^{N} L(y_j, \sum_{m=1}^{k} \alpha_m G(b_m) + \alpha_{k+1} G(b_{k+1})) \tag{5.6}$$

as a function of α_{k+1} and holding α_1, ..., α_M fixed. After M iterations, (5.6) will have the final form as in (5.5). The final prediction rule is determined by the sign of the weighted sum of these M classifiers $G(b_m), m = 1, 2, \ldots, M$:

$$\hat{G}^* = sign\left(\sum_{m=1}^{M} \alpha_m \hat{G}(b_m)\right) \tag{5.7}$$

Since the importance of identified building blocks is different with regard to the influence scores, different weights are assigned to each building block. To take the importance of the building blocks and avoid the singularity problem by stacking method, boosting method is a better choice in aggregating these building blocks. We discuss different classifiers used by boosting in the following section.

Boosting KNN classifier

Since the building blocks identified by the proposed influence score may include global or local patterns of specific forms, one of the best classifiers to use is K-nearest neighbor classifier, which takes advantage of the structure and specific high order interactions in each building block. However, directly applying the KNN classifier in boosting is not effective. The K-nearest neighbor method is a memory-based method that makes use of all training samples to predict the class label of testing set. Tuning the $\{\alpha_i, \forall i = 1, 2, \ldots, M\}$ is not feasible because, not omitting some instances, the KNN classifier will always achieve 100% training set accuracy with K=1 (i.e. the training instance help classify itself.), making boosting not feasible. Therefore, we will compute the weighted error rate by cross-validation instead of using the whole set in boosting. To best maintain the structures of identified building blocks, we use leave-one-out cross-validation during the boosting KNN algorithm to compute weighted error rate and to tune the best weight of $\{\alpha_i, \forall i = 1, 2, \ldots, M\}$. The detailed algorithm is described as follows:

Algorithm 5.1: Boosting Based on KNN classifier

Step 1: *Initialize observations with equal weights* $w_i = \frac{1}{N}$, *i=1,2,...,N with pre-selected training set;*

Step 2: *For* $m = 1$ *to* M *iterations (*M *is the number of input classifiers);*

a. Fit classifiers $G(.)$ with training set using w_i

a1. For $t = m$ to M classifiers, fit classifier $G(b_t)$ (i.e $G(.)$ is KNN)

a2. Predict $\hat{y}_t(i) = G(b_{t[-i]})$, $\forall i = 1, \ldots, N$ observations by LOOCV

where $t[-i]$ indicates that fitting with t^{th} classifier without observation i

a3. Compute

$$err_t = \begin{cases} \sum_{i=1}^{N} w_i 1(y_i \neq \hat{y}_t(i)) / \Sigma_{i=1}^{N} w_i & \Sigma_{i=1}^{N} 1(y_i \neq \hat{y}_t(i)) \neq 0 \\ 1/2N & \Sigma_{i=1}^{N} 1(y_i \neq \hat{y}_t(i)) = 0 \end{cases}$$

b. From the family of classifiers G_t, find the classifier $G(b_m)$ that minimizes the weighted error rate (err_t):

$$G(b_m) = \underset{G(b_m) \in G_t}{\operatorname{argmin}} err_t$$

c. $\alpha_m = \log \frac{1-err_m}{err_m}$

d. Set $w_i = w_i \exp [\alpha_m 1(y_i \neq \hat{y}_i(m))]$, *i*=1,2,...,*N*

Step 3: Output

$$G(x) = \begin{cases} 1 & \sum_{m=1}^{M} \alpha_m G(b_m) \geq 0 \\ -1 & Otherwise \end{cases}$$

In this algorithm, the input order of each building block is determined by the minimized weighted error rate. In general, the misclassified training sets in previous building blocks will have their weights increased; whereas, the weights are decreased

for those which have been classified correctly. After the stage of filtering independent building blocks, we notice that occasionally the building blocks achieve perfect prediction with a zero weighted error rate. For such j^{th} classifier, in Step a3, the error rate is set as $\frac{1}{2N}$. The classifier $G(b_j)$ will not affect the weight of each data point and its weight α_j is set as a constant proportional to sample size (i.e. $\alpha_j = log(2N-1)$). In step 2b, if there are ties among the input classifiers, the most informative one with higher influence score is chosen first. In general, as the building blocks are added one by one to the classification rule via the boosting algorithm, the training error is expected to decrease quickly, which would reflect an improvement of the fit to the training set. However, the testing sample error rate obtained by sequentially adding the building blocks is not guaranteed to decrease since the information of testing samples are not used to construct the prediction rule.

We have to tune the $\{\alpha_i,\ \forall i = 1, 2, \ldots, M\}$ by cross-validation in boosting algorithm with KNN classifier. Other methods that are not memory-based classifiers can skip step $a2$. We apply the other two methods, the logistic regression incorporating interactive effects in Wang et al (2012) and classification and regression tree (CART), to evaluate the performance with regard to identified building blocks. The former classifier with higher order interactions included has the capability to generate classifiers of global structure, and the latter one is able to adopt both global and local structure into the classifier dependent on the specific building blocks.

Logistic Regression Classifier

Logistic regression is a general method to evaluate performance based on global structure. Wang et al (2012) applied logistic regression to a few two-class classification problems. They generated an exhaustive model with all higher order interaction effects based on the variable modules they identified. For example, if the variable

module consists of 3 variables $\{X_1, X_2, X_3\}$, the full model is as follows:

$$ln\{\frac{P(Y=1)}{P(Y=-1)}\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_1 X_2 + \beta_5 X_1 X_3 + \beta_6 X_2 X_3 + \beta_7 X_1 X_2 X_3 \quad (5.8)$$

Where $P(Y = 1)$ is the probability of class 1 and $P(Y = -1)$ is the probability of class 2. In addition, by stepwise selection via AIC score, the submodel with lowest AIC was further used. Therefore, each input classifier in the boosting algorithm may contain different interactive effects. They further aggregated these classifiers to perform prediction on the testing set.

In Wang et al (2012), the boosting algorithm with refined logistic regression classifiers performs very well with identified variable modules; however, there are some potential drawbacks to fit an exhaustive model if the number of variables of a certain variable module is huge.

First, the model complexity grows exponentially, if the sample size is small and the variable module consists of too many variables, it is impossible to fit an exhaustive model when $n < 2^p - 1$ (where p is the number of variables in the variable module.)

Second, there is an overfitting issue when fitting the model in the training set. We observed that the model including higher order interactions is inclined to separate the training set perfectly when n is close to the number of parameters used in the model. This will lead to overfitting in prediction. Although the generalized error in testing set is controlled well when the classifiers are added in boosting algorithm, we also observe that the testing error rate of logistic regression classifier fluctuates greatly over the first few iterations.

Classification and Regression Tree

Classification and regression tree (CART) is a nonparametric method that produces either classification or regression trees, depending on whether the dependent variable

is categorical or numerical, respectively. Both recursively make binary splits based on the predictors such that at each division, the resulting two subsets of data are as homogeneous as possible. CART aims to best minimize residual deviance for each split with the response of interest. For 2-class classification tree, the outcome only takes values -1, 1. The region m is defined as R_m with N_m observations. The proportion of class 1 and class -1 in node m is defined as the following

$$\begin{aligned} p_{m(1)} &= \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = 1) \\ p_{m(-1)} &= 1 - p_{m(1)} \end{aligned}$$

The observations in node m are classified by:

$$\hat{Y}_{(R_m)} = \begin{cases} 1 & p_{m(1)} > p_{m(-1)} \\ -1 & Otherwise \end{cases}$$

In general, there are no ties in the nodes since each split will make class distribution as homogeneous as possible. Similar to the KNN method, CART is also a nonparametric and nonlinear classifier. It can capture local patterns of the data. Boosting with tree classifiers may allow one to take advantage of the patterns identified in the informative building blocks. The key disadvantage is that CART splits only by one variable at a time. If higher order interactions exist, in fitting a better model, the model complexity grows, possibly leading to overfitting

The figure 5.5 is the result of CART based on the first building block (g1904, g2737, g3481, g4055, g4705) shown in table 5.1. The tree has 4 levels and the label of the node or leaf is the predicted class if the subject satisfies the rules and is assigned to specific node. The subjects will go to the left branch if they satisfy the criteria, otherwise they will be assigned to the right branch. For example, the subject is assigned as class 1 if the expression of g4055>0.082 and g3481<-0.2225. We observe the constructed tree only involves 4 genes. Although there are 5 genes in this building block, gene 1904 may not be informative in tree classifiers where it is not used to split

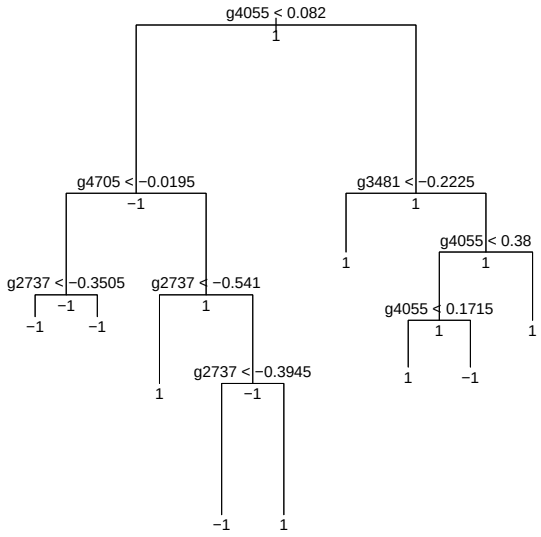

Figure 5.5: Classification tree based on the first building block identified with k=5

in the classifier. The misclassification error rate in 78 training subjects is 0.0513 (4/78).

5.3.4 Performance Evaluation

To evaluate the performance of the proposed procedure, we have to define the loss function. The typical loss function in classification is 0-1 loss function such as:

$$L(y, \hat{y}) = \begin{cases} 1 & y \neq \hat{y} \\ 0 & otherwise \end{cases}$$

where $\hat{y}$ is the prediction based on the classifiers.

If there is no independent testing set, q-fold cross-validation (q-fold CV) is used to evaluate performance. Many studies use $q = n$, the so called leave-one-out cross-validation (LOOCV). In this procedure, one uses a single sample from the original

sample set as the testing sample and builds the model based on the remaining samples from the training set. The procedure will repeat such that each sample is used exactly once as the testing sample. However, LOOCV has a few shortcomings. First, it does not perturb the data enough and will lead to higher variance in spite of the estimator being approximately unbiased for the prediction error. Second, the estimates in each fold are highly correlated. Third, the generalization error is underestimated. Zhu (2008) suggested a good compromise of $q = 5$ or 10 to avoid too much information incorporated during the training stage.

For q-fold CV ($q \neq n$), the dataset is approximately equally divided into q subsets. Each time one of the q subsets is treated as the test set and the remaining subsets are combined to form the training set. Every sample gets to be in a test set exactly once and in the training set $q-1$ times. In our procedure, besides the testing set in breast cancer, we use $q = 10$ fold cross-validation to evaluate classification performance in the three data set. For each fold, the 90% training set is used to tune our models, first screening the potential variables and obtaining the top B_2 informative and non-overlapped building blocks. Second, within the training set, we further tune the boosting parameters $\{\alpha_j,\ j{=}1,2...,B_2\}$. In general, the internal cross-validation error evaluated by training subjects will keep decreasing as more and more classifiers become involved. Therefore, the performance was evaluated by external cross-validation error rate defined as:

$$Err^{(CVE)} \quad = \quad \frac{1}{n}\Sigma_{i=1}^{n} L\left(y_i,\ sign\left(\Sigma_{j=1}^{B_2}\hat{\alpha}_j\hat{G}(b_j)\right)\right) \tag{5.9}$$

where $\{G(b_j),\ j{=}1,2...,B_2\}$ is the classifier constructed based on the building blocks identified in the screening procedure. In this thesis, we first evaluate the breast cancer data with a 19 subjects testing set with the prediction rules built by 78 subjects training set. We further evaluate the results by 10-fold cross-validation to 97 breast cancer, 62 colon cancer and 102 prostate cancer microarrays.

5.3.5 Performance on Breast Cancer Testing Set

As discussed in previous sections, 78 training subjects are used to find independent building blocks and to tune the parameters in boosting algorithm. With varied k= 1, 3, 5, we first select 300 genes with highest return frequency calculated from top 1% pairwise scores and further apply our backward elimination algorithm to find important building blocks. The parameters used in the algorithm are set as B=5,000,000 and $d = 6$ (i.e. $\lfloor log_2 n \rfloor$). We observe that tens of thousands of the building blocks have an influence score greater than 0.7. After filtering out the correlated building blocks with common genes, there are (50, 42, 35) building blocks left with screening parameters k=(1, 3, 5). We apply the boosting algorithm with KNN, refined logistic regression and CART, based on these independent building blocks. The aggregated classifier was used to predict 19 testing subjects. Table 5.2 shows the best result by the proposed procedure with different screening parameters k, minimized error rates and their corresponding numbers of building blocks used. Figure 5.6 also shows the detail performance of different classifiers using the boosting algorithm.

In figure 5.6, we observe that performance among different classifiers improves as the number of building blocks is increased, especially after about 15 of them are included. Boosting with these classifiers leads to large fluctuations in the first few steps, and the refined logistic regression classifier has the largest fluctuations. The logistic regression classifier constructed based on identified building blocks has poor prediction power in the first few steps across different k. It performs the poorest as k=1 compared to all the other classifiers. The higher error rates in the first few steps across different classifiers also indicate the difficulty of classification in breast cancer microarray. Although it becomes stable when more building blocks are included, the error rate remains high in this breast cancer data set. For example, by 1NN classifier with k=1, the error rate starts with 31.58% (i.e. 6/19) and reaches the minimized error rate 10.53% (i.e. 2/19) when 16 building blocks are included, and become stable

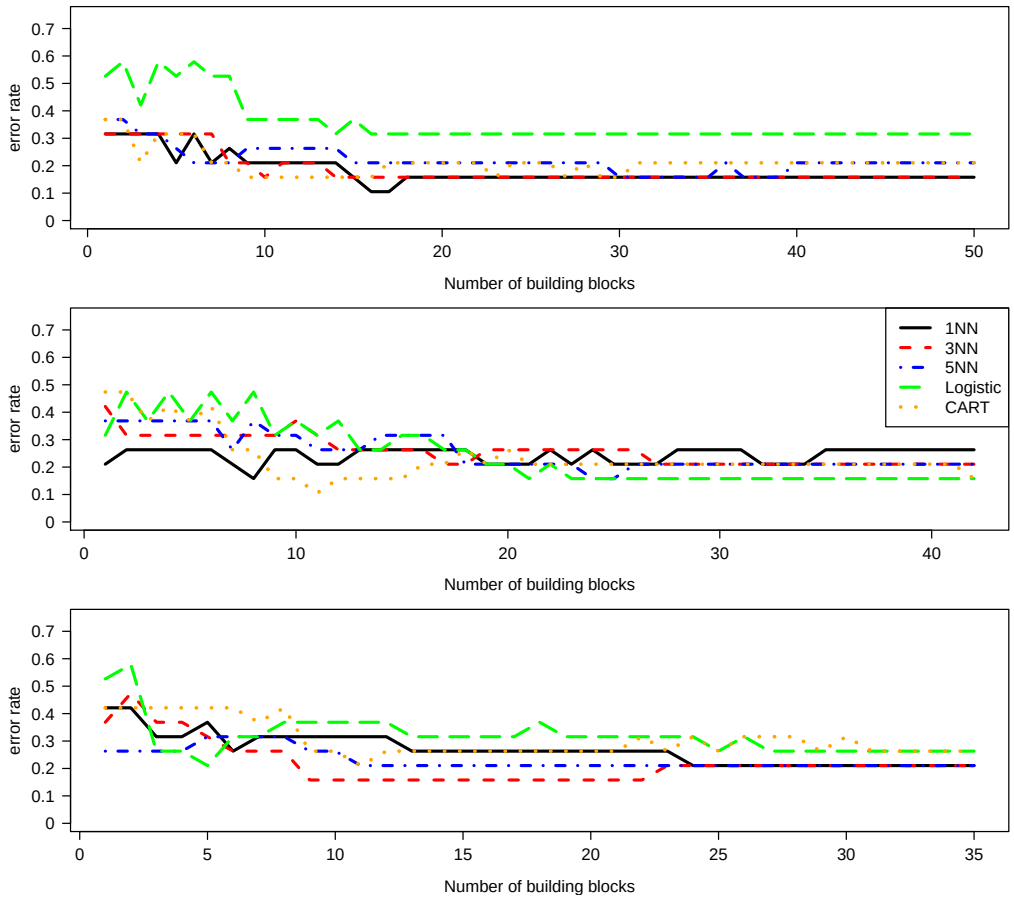

Figure 5.6: Performance of classifiers on 19 breast cancer testing set with screening parameters k=1 (upper), k=3 (middle), k=5 (bottom)

Table 5.2: The best performance for 19 breast cancer testing subjects

Screening (k)	Classifier	Error Rate	Number of building blocks
1	1NN	0.1053 (2)	16
	3NN	0.1579 (3)	10
	5NN	0.1579 (3)	30
	Logistic	0.3158 (6)	14
	CART	0.1579 (3)	9
3	1NN	0.2105 (4)	8
	3NN	0.2105 (4)	17
	5NN	0.1579 (3)	24
	Logistic	0.1579 (3)	21
	CART	0.1053 (2)	11
5	1NN	0.2105 (4)	24
	3NN	0.1579 (3)	9
	5NN	0.2632 (4)	11
	Logistic	0.2105 (4)	5
	CART	0.2105 (4)	11

with 3 wrong predictions after more than 18 clusters added.

The minimized error rate in table 5.2 is 10.53% as k=1 by 1NN with 16 building blocks and k=3 by CART classifiers when 11 building blocks are included. The best performance of the 19 subjects testing set is comparable to a many existing studies. Pochet et al (2004) applied SVM with RBF kernel and reached the minimized error rate with 31.58%. Yeung (2005) used bayesian model averaging method and obtained 15.8%. Li and Yang (2005) applied the SVM recursive feature elimination method and reached the same error rate as our best result. Wahde and Szallasi (2006) applied evolutionary algorithm to select important features and also attained 10.53% prediction error with the LDA classifier. Wang et al (2012) used a very similar concept but with discrete information measure (i.e. GTD scores) to select variable modules and applied boosting logistic regression classifier to reach a perfect error rate. However, unlike their methods, which focus on finding global pattern of variable modules such that they dichotomized each gene into 2 groups, high and

low with two means clustering algorithm, our screening methods focus more on the local structures with specific numbers of k. This might explain why, in this testing set, k=1 with boosting logistic has 31.58 % error rate, which is the worst among all other results. It may also imply the local structure k=1 is not able to cooperate well with the classifiers focusing on global structure, especially in this data set. As k increases in the screening step, the best prediction result with logistic regression classifier improves. In addition, KNN classifier with k=1 and CART classifier with k=1, 3 work well with identified building blocks in this testing set. Their error rates are all smaller than 20%, with 3 or less prediction errors among 19 testing subjects. In general, the building blocks work better with the nearest neighborhood and CART classifiers in this dataset. This result may also indicate that specific local patterns exist in breast cancer, and combine these classifiers with identified building blocks will take advantage of this situation.

5.3.6 Cross-Validation for Microarrays

In this section, we evaluate the proposed procedure with external 10-fold cross-validations on three microarrays. With similar setting in finding the independent building blocks, in the pairwise screening step, the top 1% highest pairwise scores are used to compute the return frequency and those with top 300 high returned genes are considered in the next step. The parameters in the backward elimination algorithm are set as B=5,000,000 and d=$\lfloor log_2 n \rfloor$. Due to the weak signal of breast cancer the building blocks with score greater than 0.7 are retained. For the other two microarrays, 0.8 are used as the threshold to select building blocks. We further filter the non-overlapped building blocks and the total number of them retained in the 1^{st} fold with k=5 are (23, 35, 28) for breast, colon and prostate cancers, respectively.

Table 5.3 lists the top 10 non-overlapped building blocks in the 1^{st} fold of each data set with screening parameter k=5. It shows that the strongest signals contained

in breast cancer are relatively weak compared to those identified in colon cancer and prostate cancer data. The highest score in breast cancer is only 0.8151 with 5 genes (g547, g771, g4055, g4226, g4916) forming the building block, and there are only three of them having scores greater than 0.8. As for the other two microarrays, the information contained in the top building blocks is relatively strong. Especially for prostate cancer data, the top 10 joint influence scores are all greater than 0.9.

Table 5.3: Top 10 non-overlapped building blocks of 1^{st} fold with k=5

Breast		Prostate		Colon	
Genes	Scores	Genes	Scores	Genes	Scores
547,771,4055,4226,4916	0.8151	5754,6118,10167,10605	0.9541	513,1221,1328,1346	0.9343
141,243,1601,1609	0.8124	4371,5972,9105,10225,10470	0.9486	1042,1260,1668,1843	0.9240
59,2294,3108,4504,4836	0.8031	2772,7000,7076,11804,11875	0.9388	572,1210,1400	0.9101
1763,2202,2283,3104,3315	0.7971	6648,10682,11791,12086	0.9372	1360,1597,1728,1873	0.9093
934,1331,2922,4120,4912	0.7781	4174,6841,11751,12428	0.9240	14,187,1060,1990	0.9005
2259,3381,4025,4096	0.7680	6445,8885,8932,9994	0.9232	520,1487,1582,1836	0.8908
795,1727,1897,3095,4705	0.7608	6365,8898,10610	0.9182	70,581,1466,1924	0.8903
33,1419,1615,4054,4374	0.7564	652,4215,4766,8967	0.9146	377,627,796,1465,1560	0.8897
698,1334,1345,1681,2751	0.7446	3767,5195,7453,9783	0.9123	493,889,1233,1380	0.8868
98,3685,4489	0.7368	6395,6814,10071,11414	0.9063	32,279,365,732,1504	0.8860

The number of non-overlapped building blocks varies in different folds with the same thresholds. Table 5.4 lists the summary statistics of the number of non-overlapped building blocks identified based on their corresponding thresholds.

Table 5.4: Summary of non-overlapped building blocks in 10-fold cross-validation

	Breast			Prostate			Colon		
Screening (k)	*1*	*3*	*5*	*1*	*3*	*5*	*1*	*3*	*5*
Max	47	43	39	47	41	36	51	46	41
Median	40	34	28	42.5	38	34.5	46	41	34.5
Min	32	27	23	39	34	31	40	33	28
AvgSize	4.054	4.217	4.419	4.018	4.121	4.443	3.602	3.839	3.969
SDSize	0.128	0.105	0.139	0.048	0.091	0.083	0.061	0.075	0.089

AvgSize: Average number of genes included in building block

SDSize: Standard deviation of genes included in building block

We observe that the number of non-overlapped building blocks decreases as the screening parameter k increases. With k=5, the numbers are overall less than those identified with smaller k. Furthermore, the average size of genes included in each building block increases as k gets larger. In breast cancer, the standard deviation of the genes involved in each building block is larger than that in the other two data sets.

We finally construct classifiers from these non-overlapped building blocks identified in each fold and aggregate them to form a final prediction rule. The results are listed in table 5.5. In addition, we apply our influence score and Pearson's correlation to screen their corresponding top 50 informative variables marginally. Similarly, we use the same classifiers with individual genes and further aggregate them by the boosting algorithm. Table 5.6 and table 5.7 list the best results of these two gene sets.

Table 5.5: The best performance of 10-fold cross-validation by proposed procedure

Screening (k)	Classifier	Breast	Prostate	Colon
1	1NN	0.1856 (18)	0.0686 (7)	0.1129 (7)
	3NN	0.2371 (23)	0.0588 (6)	0.0806 (5)
	5NN	0.2062 (20)	0.0588 (6)	0.0968 (6)
	Logistic	0.2245 (22)	0.0588 (6)	0.0806 (5)
	CART	0.2245 (22)	0.0686 (7)	0.1129 (7)
3	1NN	0.1546 (15)	0.0392 (4)	0.0968 (6)
	3NN	0.2371 (23)	0.0392 (4)	0.1129 (7)
	5NN	0.2371 (23)	0.0490 (5)	0.1129 (7)
	Logistic	0.2474 (24)	0.0490 (5)	0.0968 (6)
	CART	0.2474 (24)	0.0686 (7)	0.0806 (5)
5	1NN	0.2061 (20)	0.0196 (2)	0.0968 (6)
	3NN	0.1959 (19)	0.0294 (3)	0.0806 (5)
	5NN	0.1856 (18)	0.0392 (4)	0.0645 (4)
	Logistic	0.1959 (19)	0.0490 (5)	0.0645 (4)
	CART	0.1959 (19)	0.0588 (6)	0.0968 (6)

Sample size: Breast Cancer (97), Prostate Cancer (102), Colon Cancer (62)

Logistic: Logistic regression model is constructed by the same procedure as Wang et al (2012).

Table 5.6: The best performance of 10-fold cross-validation with top 50 genes selected by marginal influence scores

Screening (k)	Classifier	Breast	Prostate	Colon
1	1NN	0.1959 (19)	0.0588 (6)	0.1613 (10)
	3NN	0.1959 (19)	0.0588 (6)	0.1129 (7)
	5NN	0.2371 (23)	0.0588 (6)	0.1613 (10)
	Logistic	0.2268 (22)	0.0490 (5)	0.1129 (7)
	CART	0.2268 (22)	0.0686 (7)	0.1452 (9)
3	1NN	0.1856 (18)	0.0490 (5)	0.1290 (8)
	3NN	0.2268 (22)	0.0588 (6)	0.1129 (7)
	5NN	0.2784 (27)	0.0490 (5)	0.1290 (8)
	Logistic	0.1959 (22)	0.0392 (4)	0.1290 (8)
	CART	0.2474 (24)	0.0686 (7)	0.1613(10)
5	1NN	0.1856 (18)	0.0490 (5)	0.1129 (7)
	3NN	0.2165 (21)	0.0392 (4)	0.1129 (7)
	5NN	0.2990 (29)	0.0490 (5)	0.1613 (10)
	Logistic	0.1959 (19)	0.0392 (4)	0.1290 (8)
	CART	0.2886 (28)	0.0686 (7)	0.1452 (9)

Table 5.7: The best performance of 10-fold cross-validation with top 50 genes selected by absolute Pearson's correlation

Classifier	Breast	Prostate	Colon
1NN	0.2784 (27)	0.0882 (9)	0.1774 (11)
3NN	0.2886 (28)	0.0686 (7)	0.1452 (9)
5NN	0.3093 (30)	0.0686 (7)	0.1290 (8)
Logistic	0.2680 (26)	0.0588 (6)	0.1452 (9)
CART	0.2784 (27)	0.0784 (8)	0.1613 (10)

Breast Cancer

In table 5.5, the error rate in breast cancer is relatively high which implies the difficulty in classification of this microarray. The result with 1NN classifier attains 15.46% error rate when screening with k=3. As k=5, 3NN, 5NN, logistic and CART classifiers have accuracies all higher than 80%. The accuracies with building blocks identified by k=5 are more stable than those with k=1, 3. Figure 5.7 shows the cross-validation

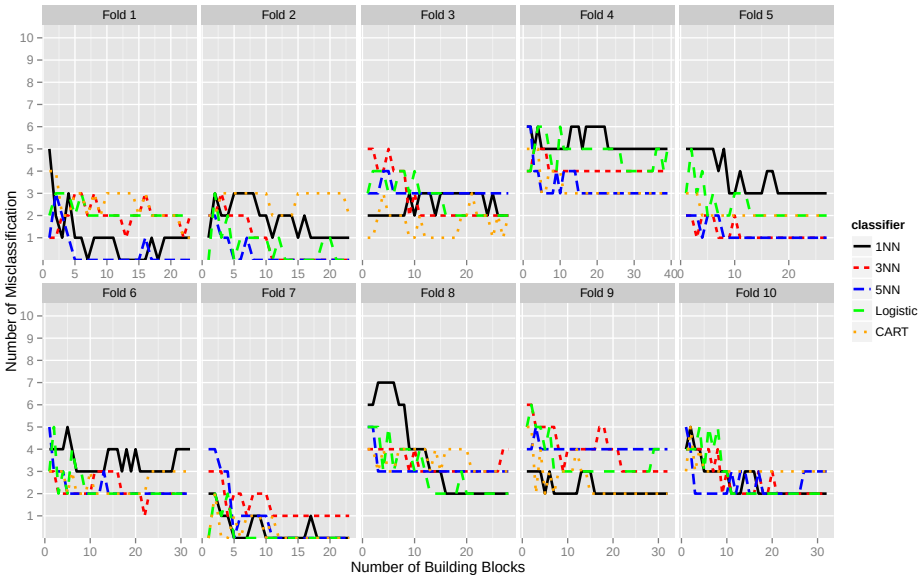

Figure 5.7: 10-fold cross-validation for breast cancer with k=5 : misclassifications vs. number of building blocks

with screening parameter k=5. These classifiers behave differently in each fold and most of the folds are challenging to predict correctly. In general, we observe there are large fluctuations if only a few building blocks included. As the number of the building blocks increases, the performances improve and approach stability although the number of misclassified subjects in some folds is still very high. For example, in fold 4, all the classifiers have at least 3 misclassified subjects and 1NN has half of the subjects misclassified. We also observe the results with 10-15 building blocks by 5NN classifier are relative stable and have better performance in this dataset.

In table 5.6, boosting with genes selected by marginal influence scores, the minimized error rates is 18.56% by 1NN classifier with k=3, 5 but 5NN and CART have higher error rates. When k=1, the performances are relatively stable compared to that with k=3, 5. The results with top 50 strongest genes screened by Pearson's correlation are shown in table 5.7, the error rates by these aggregated classifier are consistently high. The logistic regression classifier reached 26.8% error rate and other

classifiers are even higher. From these results, we observe the performances of the building blocks identified by k=5 are more stable and the data may have specific non-linear effects leading to the higher error rate by genes with high Pearson's correlation.

Many classification methods are applied to this data and most of them do not have attractive results. In table 5.8, many studies select features with filtering methods such as Pearson's correlation (Van't Veer, 2002), signal-to-noise ratio (Peng, 2005) and F-ratio (Diaz-Uriarte & de Anres, 2007). Their error rates are about 20% to 35%. Other studies have significant results. Song et al (2007) applied SVMRFE and reached error rate of only 7.7%, but they used full set to do feature selection which may have lead to an optimistic result. Wang et al (2012) used logistic regression based on variable modules and reached 8% but they use 10 random sampling which may have potential advantage that the chance of selecting difficult samples is small. Our best performance 15.46% by building blocks is also among the top list.

Table 5.8: Comparisons with other existing methods of breast cancer data set

Author	Feature Selection	Classifier	CV	Min Error(%)
Van't Veer et al. (2002)	Correlation	Correlation	LOO	27
Peng (2005)	SNR	SVM	LOO	24.7
	SNR	Bagging SVM	LOO	21.6
	SNR	Boosting SVM	LOO	21.6
	SNR	Ensemble SVM	LOO	18.6
Alexe et al (2006)	LAD	LAD	CV	18.3
Diaz-Uriarte & de Anres		Random Forest	bootstrap	34.2
(2007)	F-ratio	SVM	bootstrap	32.5
	F-ratio	KNN	bootstrap	33.7
Zhu et al. (2007)	RFE	SVM	10-fold	29
Hewett et al.	MDR*	MDR	10-fold	37.11
Ng (2010)	Clustering	LR	LOO	28.2
Wang et al (2012)	Retention Frequency	Boosting logistic	10-fold rCV	**8**
Huang and Lo	I score (k=3)	1NN	10-fold	**15.46**
	I score (k=5)	5NN	10-fold	18.56

LAD:Logical analysis of data; **SNR**:Singal-to-noise ratio; **MDR**:Multi-Dimension Ranker;

Prostate Cancer

The prostate cancer microarray is an easier dataset for classification compared to breast and colon cancers since the error rate in table 5.5 are low. With k equal to 3 or 5 in screening step, most of the error rates by nearest neighbor and logistic regression classifiers with interactive effects in our proposed procedure are less than 5%. The minimized error rate is 1.96% by 1NN classifier with k=5. The accuracies of boosting with CART are slightly lower but all of them are better than 95% with different corresponding k.

Figure 5.8 shows the 10-fold cross-validation result with k=5 in the screening step. The 1NN classifier performs very well in prostate cancer with identified building blocks. There are only 2 subjects never predicted correctly in fold 5 and 7. In fold 2, there are 4 initially misclassified subjects, but all are correctly classified after 8 building blocks have been included.

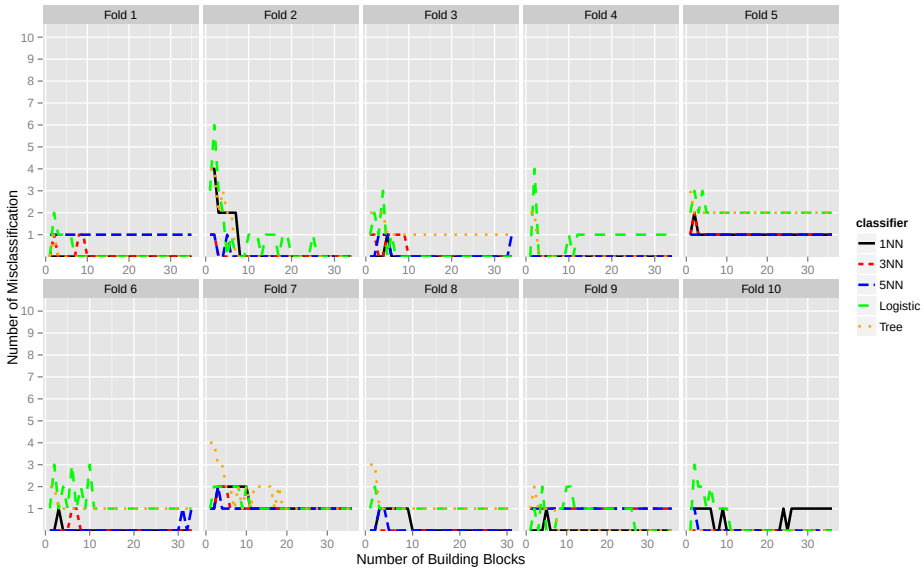

Figure 5.8: 10-fold cross-validation for prostate cancer with k=5 : misclassifications vs. number of building blocks

We also observe that the best predicted accuracy appears as 10-20 building blocks

involved by nearest neighborhood classifiers. The logistic regression classifier exhibits large fluctuations in the first few steps of many folds, and this might imply the input building blocks contain local signals, so that the logistic classifiers have diverse predictions in the beginning and the results improves as the number of building blocks increases. In general, the number of misclassified subjects decreases as the number of building blocks increases.The predicted results become stable when 10 to 20 building blocks are involved.

In table 5.6, the genes are selected with marginal influence measure, and the error rates are also around 5% but slightly higher than the error rates by building blocks. In table 5.7, the logistic regression classifier reached 5.88% error rate and the nearest neighborhood classifier performs slightly better than that in Singh et al (2002). They also used Pearson's correlation to select important genes and reach 8% error rate with LOOCV by nearest neighbor method. From table 5.5, 5.6 and 5.7, the proposed measure has slightly better performance if we includes the interactive effects. This may indicate that the local and interactive effects exist in the identified building blocks, leading to enhanced predictive power, which is not captured by the genes selected by linear screening method.

Many variable selection methods are used with different classifiers to evaluate the performance in prostate cancers as in table 5.9. Those methods explored the joint interactions among genes and most of them performed well. For example, Dettling and Buhlmann (2003) used wilcoxon statistics as a criterion to identify clusters of variables. With 15 clusters of variables, they reached error rate of 4.9%. Wang et al (2013) found top score gene group based on chi-square statistics and Tan et al (2005) applied a rank based method to identified top informative pairs of genes. Both of them attained 4.9% LOOCV.

Zhang et al (2012) applied binary matrix shuffling filter to find potential interactions with SVM classifier and attained 3.24% 10-fold cross-validation error rate.

Dagliyan et al (2010) applied information gain to do pre-screen and by solving an optimization problem (hyper-box enclosure method) to find interactive effects among genes and reached error rate less than 4%. Other studies not considering interaction in the data set have higher error rate. For example, Liu et al (2011) had 13.82% with 10NN, Statnikov (2005) reached 8% by SVM, and Dagliyan had 7.84% error rate with logistic regression classifier.

Table 5.9: Comparisons with other existing methods of prostate cancer data set

Author	Feature Selection	Classifier	CV	Min Error(%)
Singh et al. (2002)	correlation	KNN	LOO	8.00
Dettling &	Wilcoxon	1NN	LOO	4.9
Buhlmann(2003)		Trees	LOO	5.88
Tan et al.		TSP *	LOO	4.9
(2005)		k-TSP *	LOO	8.82
		*k*NN	LOO	23.53
Statnikov et al.		SVM	10 folds	8.00
(2005)				
Kucukural et al.	GA*	SVM	10-fold	3.92
(2007)				
Hewett et al. (2008)	MDR*	MDR	10-fold	11.76
Ahdesmaki &	CAT score*	LDA	10-fold	7.07
Strimmer (2010)		DDA	10-fold	4.97
Dagliyan et al.	Information gain	HBE*	LOO	3.92
(2011)		LR	LOO	7.84
		RF	LOO	5.88
Liu et al.	SVM-RBF-RFE	5NN	10-fold	14.82
(2011)		10NN		13.82
Zhang et al.	BMSF*	SVM	10-fold	3.24
(2012)	BMSF	NB	10-fold	10.4
	BMSF	LDA	10-fold	4.51
Wang et al.		Chi-TSG*	5-fold	9.8
(2013)			LOO	4.9
Huang and Lo	I score (k=5)	1NN	10-fold	**1.96**
		3NN	10-fold	**2.94**

BMSF: Binary Matrix Shuffling Filter; **CAT score**: correlation adjusted t score;

Chi-TSG: Chi-square top scoring genes; **GA**:Genetic Algorithm; **HBE**: hyper-box enclosure method;

k-TSP: k - top scoring pairs; **MDR**:Multi-Dimension Ranker; **TSP**: Top scoring pairs;

Colon Cancer

In table 5.5, our procedures achieved around 90% accuracy with different combinations of screening parameters k and classifiers. Screening with k=5, boosting with both 5NN and logistic regression have the minimized error rate 6.45%. The best result with CART is 8.06% error rate when k=3.

Figure 5.9 shows the detailed cross-validation result with k=5. We observe the number of misclassifications decreases as more building blocks are added across all folds. There are 4 subjects in folds 4, 5, 8 and 10 that are difficult to be predicted accurately by 5NN classifier. As for logistic regression classifier, one subject in folds 4, 8 and two subjects in fold 10 are not classified correctly. In addition, with at least 20 building blocks included in each fold, both 5NN and logistic regression classifier have the best performances among all other classifiers.

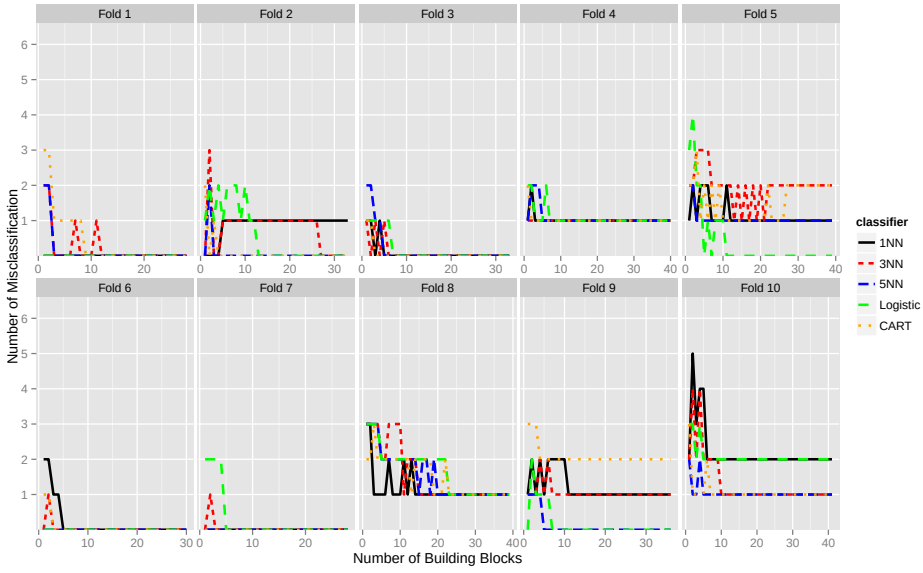

Figure 5.9: 10-fold cross-validation for colon cancer with k=5 : misclassifications vs. number of building blocks

On the contrary, the results with marginal signals in table 5.6 and 5.7 show that the prediction accuracies deteriorate generally compared to the results with building

blocks. All the error rates are greater than 10%. In colon cancer, boosting with building blocks having better results that might be due to the existence of higher order interactions that doing prediction based on important marginal variables may lose important interactions among the genes.

Table 5.10: Comparisons with other existing methods of colon cancer data set

Author	Feature Selection	Classifier	CV	Min Error(%)
Ben-Dor et al. (2000)	TNoM scores	Clustering	LOO	11.3
		KNN	LOO	19.4
		Boosting	LOO	27.4
Furey et al. (2000)	Globe	SVM	LOO	9.68
Zhang et al.(2001)	Purity	Decision Tree	5-fold	**6.45**
Dettling & Buhlmann (2003)	Wilcoxon	1NN	LOO	16.13
		Trees	LOO	16.13
Lee et al.(2003)	Soft-thresholding	SVM(RBF)	3-fold	12
	Wilcoxon	kNN(5)	3-fold	13
Liu et al. (2004)	Ranksum	Ensemble NN	LOO	8.06
	PCA		10-fold	9.68
Tan et al. (2005)		TSP *	LOO	8.9
		k-TSP*	LOO	9.7
		kNN	LOO	25.81
Zhang et al.(2007)	BBF*	5NN	LOO	9.68
	BBF	SVM	LOO	12.90
Alladi et al. (2008)	t	LR	10-fold	21.82
		NN	10-fold	19.09
		SVM(RBF)	10-fold	14.55
Wang et al. (2013)	CV	Chi-TSG*	5-fold	15.2
			LOO	**6.45**
Huang and Lo	I score (k=5)	5NN	10-fold	**6.45**
		LR	10-fold	**6.45**

BBF:Based Bayes error Filter; **NN**: Neural network; **TNoM**: Threshold number of missclassification;

Table 5.10 lists many different methods applied to prediction of the disease status in colon cancer. Filtering important features by marginal methods such as Ben-Dor et al (2000), Furey et al (2000), Zhang et al (2007) and Alladi (2008), result in the performances having error rates higher than 9.68%. Zhang el al(2001) applied a decision tree method with 5-fold cross-validation and reached 6.45% error rate.

However, they applied the method only the steps that occurred after selection of the informative genes. In other words, the full dataset was used to identify the informative genes, which leads to optimistic performance since the testing information is included in feature selection during training stage. Wang et al (2013) takes advantage of top informative genes and their joint interactions. It works well in prediction with error rate 6.45% in LOOCV, but the 5-fold cross-validation are relatively weak (15.2%). Dettling and Buhlmann (2003) used 10 clusters of genes and reached the leave-one-out error rate at 16.13% with both nearest neighbor and tree methods. We observe most of the method taking interaction effects into consideration perform well with error rate less than 10% (Zhang et al (2003), Tan et al (2005) and Wang et al (2013)). Our method also takes advantage of interactive effects by applying I scores (k=5) to screen important building blocks outperforms most of the studies in accuracy.

From the above results, there are many advantages of our proposed measures and framework. First, as the screening parameter k increases the building block may capture more useful information, so that the performance in the 10-fold cross-validation is better as k=5 compared to that with k=1,3. Second, the performances improve and become stable as more and more informative building blocks are added. Third, taking higher order interactions into consideration will benefit prediction accuracy. Ultimately, the microarray studies demonstrate the applicability of the proposed screening method, which works well in many different classifiers with identified important building blocks.

5.3.7 Variable Relevance

The proposed measure is evaluated on breast cancer, prostate cancer and colon cancer data sets. Due to the large number of features in these gene expression datasets, a two-stage analysis is adopted. On the first stage, we use an interaction-based pairwise screening with proposed influence measures. The top 300 highest return frequency genes in each fold were chosen to advance the second stage. In the second stage, we applied backward elimination algorithm with genes random selected from the refined variable subsets. After filtering out the non-overlapped building blocks from the top informative modules, the gene relevance is evaluated by calculating the average score of i^{th} gene from the folds:

$$Relevance\ of\ i^{th}\ gene = \frac{1}{q} \sum_{\substack{l=1 \\ i \in b_{lj}, j=\{1,2,..,B_2\}}}^{q} I(b_{lj}) \tag{5.10}$$

Where the $I(b_{lj})$ is the score of j^{th} building block in l^{th} fold that gene i belongs to. The relevance score considers both the number of returns in each fold and the influence score of building block that i^{th}gene belong to. It is possible that some genes show significant impacts in only a few folds that might be due to random splits. We should not just compute the relevance score of i^{th} gene from the folds it returned as that will lead to a biased evaluation of such gene. The relevance score aims to have the important genes returned consistently no matter what the samples were split. Therefore, with higher score of the variable relevance, the higher number of returns and score of the gene in the identified building blocks of each fold.

Breast Cancer

In breast cancer, we observe that only the top 12 genes were consistently returned in all the 10 folds and only gene 4836 has relevance score greater than 0.8. Gene 4836 is an example of having the weaker marginal information, but it shows importance by its relevance score. Its marginal information is not strong but it is rank 1 in gene

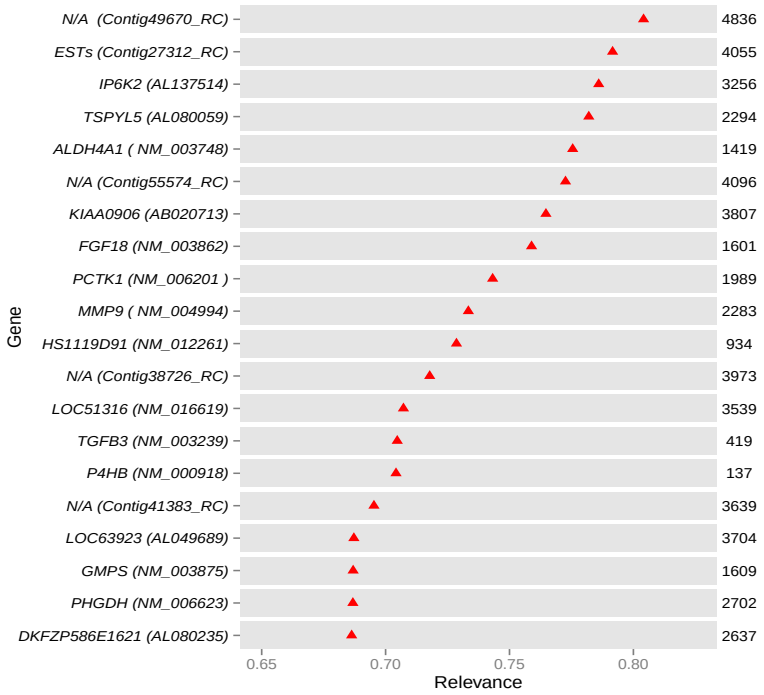

Figure 5.10: Top 20 relevant genes in breast cancer with k=5

relevance. That indicates the strong interaction effects of Gene 4836 and other genes. Figure 5.10 shows the top list of gene relevance, gene/systematic names and their corresponding variable id's. We found many of the high relevance genes have been studied in relation to breast cancer.

TGFβ family belongs to signaling pathway (KEGG) which, acts as a suppressor of primary tumor initiation during early and late stage of tumourigenicity, is known to regulate many cellular processes involved in carcinogenesis (Blobe 2000 and Itoh 2012). Hoshino et al (2011) demonstrated this in highly metastatic breast cancer cells from which *TGF-β1* and *TGF-β3* (NM003239) are abundantly expressed. Immunostaining for *TGF-β3* was inversely correlated with survival and the expression of *TGF-β3* in breast cancer tumors was shown as an independent predictor of overall survival (Ghellal A et al, 2000). The fibroblast growth factors (*FGFs*) play key roles in controlling tissue growth, morphogenesis, and repair in animals. No direct study

shows the *FGF18* (NM003875) has any relationship with breast cancer; however, it has been recently identified as an abnormal expressed gene within an expression signature predicting poor rate of survival in patients with ovarian cancers (Wei et al, 2012). *MMP9* (NM004994) involved in cancer invasion and metastasis, has been extensively explored and deemed relevant to breast cancer. The expression of *MMP9* was a prognostic marker in node-negative breast cancer (Scorilas et al., 2001) and mailgnant breast tumors increase *MM9* activity compared to benign breast tumors (Hanemaaijer et al., 2000). It might be associated with breast cancer development and tumor progression (Khrmann, 2009). *MM9* levels have positive correlation with metastatic disease and reduced relapse-free survival in patients with breast cancer (Vizoso et al. 2007; Wu et al. 2008). A recent study showed *PHDGG* (NM006623) is in a genomic region of recurrent copy number gain in breast cancer and *PHGDH* protein levels are elevated in 70 % of *ER*-negative breast cancers (Possemato et al, 2011). It also suggested that targeting the serine synthesis pathway may be therapeutically valuable in breast cancers with elevated *PHGDH* expression.

Prostate Cancer

In figure 5.11, we observe that all the gene relevance scores of the top list all very high (> 0.9) and all of them show the importance in 10 folds.

Many genes in the list are related to prostate cancer in existing literatures. For example, *MAF* plays a role in pathways of tumorigenesis (Sharad et al, 2011) and has a tumor suppressor role because it participates in *TP53*-mediated cell death (Hale et al., 2000). *HPN* is shown to be associated with prostate cancer (Burmester et al, 2004) and can be used as early detection of prostate cancer (Kimberly et al, 2008). From the genomewide studies, it is shown to be an important gene related with prostate cancer both in European (Pal et al, 2006) and Korea men (Kim et al, 2012). *ERG* is extensively explored and considered a prostate cancer biomarker. Over-expression

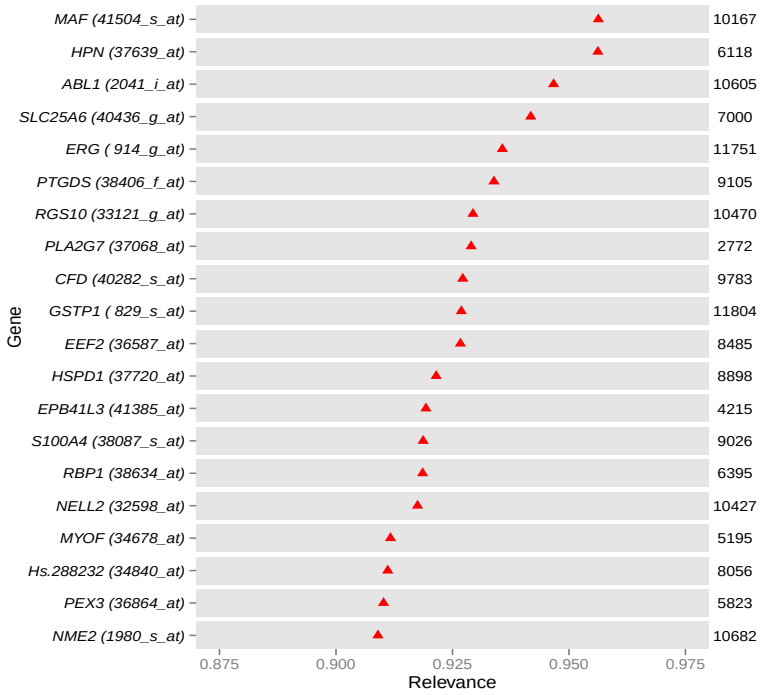

Figure 5.11: Top 20 relevant genes in prostate cancer with k=5

of *ERG* has also been considered as a specific prostate cancer biomarker which is rarely found in other cancers or normal tissues (Demichelis et al, 2007). *TMPRSS2-ERG* gene fusion is the most common variant observed in about 50% of all prostate-specific antigen screened prostate cancer patients in the United States (Mosquera et al, 2009), (Tomlins et al, 2009). *PTGDS* was one of three genes that expressed at consistently lower levels in prostate cancer compared to normal patients (Thompson, 2012). *RGS10* has been shown to be related with human colon (Lu et al, 2008) and ovarian (Hooks et al, 2010) cancers, but no direct evidence demonstrates its influence on prostate cancer. *PLA2G7*, one of the members in arachidonic acid pathway, is considered an important biomarker in 50% of prostate cancers and associates with aggressive disease (Vainio, 2011). *GSTP1* DNA methylation and protein expression status is correlated with DNA methyltransferase inhibitors treatment response in prostate cancer cells (Chiam 2011) and it is a reliable molecular biomaker for early

detection of prostate cancer among Egyptians with 90.9% sensitivity (Essawi, 2010). *HSPD1* was found to be associated with prostate cancer risk by RT-PCR (Hu, 2013). *EPB41L3* plays an important role in tumor progression including prostate cancer and the potential therapy to upregulate *EPB41L3* gene expression in prostate cancer cells are currently being developed(Bernkopf, 2008). *S100A4* is a member of S100 family of calcium-binding proteins that is directly involved in tumor metastasis (Garrett 2006). It is overexpressed during the progression of prostate cancer and could be a novel therapeutic target for human prostate cancer treatment (Saleem, 2006).

Colon Cancer

The top 20 relevant genes in colon cancer show consistent importance in 10-fold cross-validation. We observe that of the highest relevance genes as shown in figure 5.12, 15 of them have relevance score greater than 0.9. Many of them have been explored and appropriate experiments on their biological relations with colon cancer have been undertaken.

HIVEP2 (R39209) has been implicated in the regulation of immune responses and cellular proliferation (Fujii el al, 2005), and was found to be lower in cells transduced with the miR-155, which is expressed at elevated levels in human diseases including lung, breast, colon cancers (Yin et al, 2010); Gelsolin Precursor (H06524) helps maintain the integrity of cell cytoskeleton (Sun, 1999), and was found to downregulate in several tumors and its abnormal expression is among the most common defects found in human bladder and colon cancer (Porter et al, 1993; Rao, 2002); By northern blot hybridization, Hill et al (1995) revealed a high level expression of the *GUCA2B* gene in human colon and indicated a pivotal role in cGMP-mediated functions of the colon. In addition, *GUCA2B* (Z50753) (uroguanylin) is an endogenous activator of the guanylate cyclase-2C receptor, and it could be used as a non-invasive biomarker for the early detection of colorectal cancer (Liu et al, 2009); *COL1A2* (H08393) is

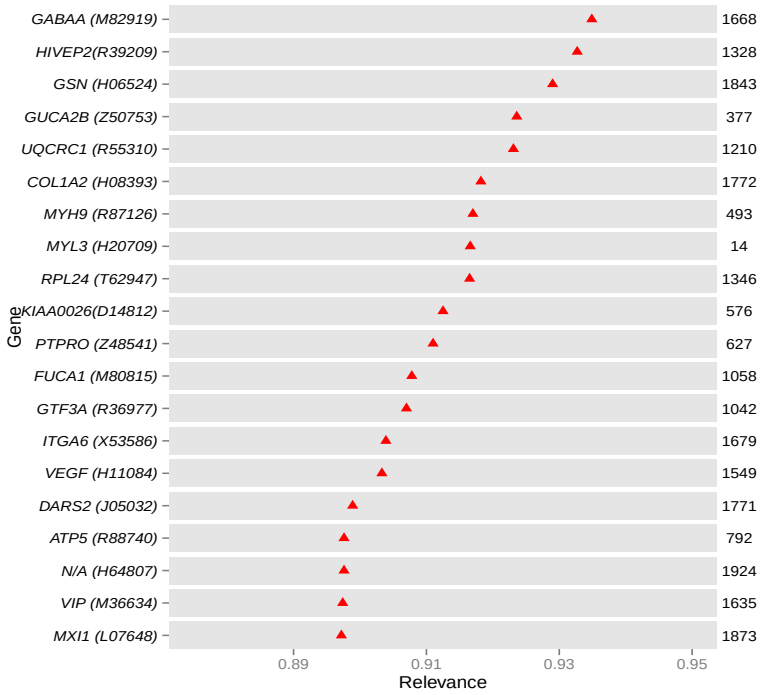

Figure 5.12: Top 20 relevant genes in colon cancer with k=5

a collagen alpha 2 chain which is involved in cell adhesion, and collagen degrading activity have been shown as part of the metastatic process for colon carcinoma cells (Guyon et al 2002; Karakiulakis 1997). *RPL24* (T62947) may play a role in controlling cell growth and proliferation through the selective translation of particular classes of mRNA (Guyon et al 2002). These two genes of colon cancer biomarkers, *COL1A2* and *RPL24*, had been applied for United States patent whose number is 20050165556 in 2005; Vasoactive intestinal peptide (*VIP*, M36634) is indicated to promote the growth and proliferation of tumor cells and patients with colorectal cancer inclined to have a higher serum level of *VIP* and a higher density of *VIP* receptors in cancer cells (Hejna et al., 2001). *MXI1* (L07648), MAX interacting protein 1, which decreased 2-fold in the colorectal cancer samples, is an antagonist of c-myc oncogene. Down-regulation of *MXI1* further is likely to enhance the activity of MYC, which was observed to be overexpressed in the colorectal tumors (Zervos, 1993).

Chapter 6

Discussion and Conclusion

In this thesis, inspired by Lo and Zheng's initiative work (2002), we propose a novel and heuristic variable selection measure based on nearest neighborhood information. The nonparametric measure has many good properties, while not being restricted to the assumption of linearity. In addition, the measure can identify informative patterns in low dimensional variable subspaces and capture high order interaction at the same time. As discussed, the genetic diseases were affected by many functional pathways (i.e. group of genes). Interactions, especially the epistasis, come in various forms. The proposed influence measure I is flexible to accommodate groups of variables and evaluate their joint association with responses making it ideal for gene expression data analysis.

The proposed measure has advantage of capturing continuous predictors compared to the original categorical score (Zheng, 2006). To apply the categorical influence measure in continuous predictors, the first step is to discretize the predictors by specific quantile or by clustering. Based on the same simulated data set in table 4.1, we dichotomize the predictors by the mean of each variable and the history of the eliminating procedure is shown in table 6.1. We observe that when either one of the influential variable is not selected, the influential variables are not have strong signal

any more even there is strong non-linear marginal effects. On the contrary, proposed continous measure is able to capture and retain the influential variables as in table 4.1.

Table 6.1: History of the eliminating procedure for four cases with categorical I score

Initial set:{1,2,3,4,5,6,7}							
Influence before drop	1.7614	2.5927	3.8500	6.6239	10.4370	20.2258	0.5268
Dropped variable	6	3	7	4	5	1	2
Initial set:{1,3,4,5,6,7}							
Influence before drop	0.9759	1.2073	1.2944	1.1753	1.002	1.1127	
Dropped variable	1	4	6	3	5	7	
Initial set:{2,3,4,5,6,7}							
Influence before drop	1.2036	1.4604	1.5069	1.8810	0.8297	0.5268	
Dropped variable	6	3	7	4	5	2	
Initial set:{3,4,5,6,7}							
Influence before drop	1.2073	1.2944	1.1753	1.0016	1.1127		
Dropped variable	4	6	3	5	7		

As for other simulation studies, with various value k, the high order information will be detected by our measures no matter what the relations are. In addition to assign the k arbitrarily, we can also apply the cross-validation method to find a suitable k. However, to reduce the computation burden and improve the variable selection procedure, moderate value of k is enough to identify importan variables. In simulation studies, we found the influential variables are always included in the returned set with different k. The microarray studies also demonstrate its capability to identify relevant genes to different kinds of complex diseases. Furthermore, various forms of joint effects among variables are able to be captured.

We also proposed a new procedure to do classification in gene expression microarrays. To reduce the computational complexity, a two stage analysis is adopted. We first screen and detect potentially important genes by interaction-based screening.

By return frequency, those genes that consistently appear with jointly high influence scores have higher potential to form influential building blocks. Secondly, the informative building blocks are generated based on a computationally intensive method, the backward elimination algorithm. In the algorithm, the repeat time B has to be set as a large number to ensure the ability to find true and informative interactions. Although the heavy burden of computations are needed in this step, taking advantage of high performance computing cluster (i.e. parallel programming environment) makes the step manageable. With the growing of advanced technology such as graphics processing unit (GPU) with thousands of cores, the computational burden in backward elimination algorithm will be further eased in the future. Based on the two stage analysis, we finally construct many different classifiers by the identified informative building blocks, and combine them into a final classification rule by boosting algorithm. The performance of proposed procedure that incorporates higher interactive effects to do classification is strong compared to many existing methods. In addition, it also outperforms the results by boosting algorithm with strong marginal effect genes selected by Pearson's correlation (i.e. assume linear relationship) and those identified by proposed influence score. That may imply the interactive effects play a role in gene expression data.

The proposed influence score is one way to evaluate the association between a set of variables and response. An alternative influence score similar to (2.4) defined as follows:

$$I = \frac{\Sigma_{i=1}^{n} k(\hat{Y}_i - \bar{Y})^2}{n\sigma_Y^2}, \tag{6.1}$$

where $\hat{Y}_i$ is the average of Y_i and its (k-1) nearest neighbors. However, unlike the influence score of (2.4) used with categorical predictors that every observation is counted exactly once, (6.1) may have observations assigned to more than one partitions. The detailed properties of (6.1) are still need to be explored.

In addition, we use Euclidean distance to evaluate the similarity of observations.

There are still many distance measurements can be explored to evaluate how the observations cluster together in the variable space. For example, Pearson's correlation to cluster together observations with similar behaviors, Spearman correlation cluster together observations whose profiles have similar shapes or show similar general trends and cosine similarity measures the cosine of the angle between two vectors of an inner product space which is popular in text mining. These different similarity measures may provide a different perspective to apply our influence scores in diverse research areas.

Bibliography

Alladi, S. M., P, S. S., Ravi, V., and Murthy, U. S. (2008). Colon cancer prediction with genetic profiles using intelligent techniques. *Bioinformation,* 3(2), 130-133.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A.J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA,* 96, 6745-6750.

Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M., and Yakhini, Z. (2000). Tissue classification with gene expression profiles. *J Comput Biol,* 7(3-4), 559-83.

Bernkopf, D. B. and Williams, E. D.(2008). Potential role of EPB41L3 (protein 4.1B/Dal-1) as a target for treatment of advanced prostate cancer. *Expert Opin Ther Targets,* 12(7), 845-53.

Blobe, G. C., Schiemann, W. P., and Lodish, H. F. (2000). Mechanisms of disease: role of transforming growth factor (β) in human disease. *N Engl J Med,* 342, 1350-8.

Breiman, L. (1998). Arcing Classifier. *The Annals of Statistics,* 26(3), 801-849.

Buhlmann, P. and van de Geer, S. (2011). Statistics for High-Dimensional Data: Methods, Theory and Applications.. *Springer-Verlag.*

Burmester, J. K., Suarez, B. K., Lin, J. H., Jin, C. H., Miller, R. D., Zhang, K. Q., Salzman, S. A., Reding, D. J., and Catalona, W. J. (2004). Analysis of candidate genes for prostate cancer. *Hum Hered,* 57(4), 172-8.

Chernoff, H., Lo, S-H., and Zheng T. (2009). Discovering Influential Variables: A Method of Partitions. *The Annals of Applied Statistics,* 3, 1335-1369.

Chiam. K., Centenera, M. M., Butler, L. M., Tilley, W. D., and Bianco-Miotto, T. (2011). GSTP1 DNA Methylation and Expression Status Is Indicative of 5-aza-2-Deoxycytidine Efficacy in Human Prostate Cancer Cells. *PLoS ONE,* 6(9):e25634.

Cover, T. and Hart P. (1967). Nearest Neighbor Pattern Classification. *IEEE Transaction on Information Theory,* IT-13, 21-27.

Demichelis, F., Fall, K., Perner, S et al. (2007). TMPRSS2:ERG gene fusion associated with lethal prostate cancer in a watchful waiting cohort. *Oncogene,* 26, 4596-4599.

Dettling, M. and Buhlmann P.B. (2002). Supervised clustering of genes. *Genome Biology,* 3(12)

Dettling, M. and Buhlmann P.B. (2004). Finding predictive gene groups from microarray data. *Journal of Multivariate Analysis,* 90, 106-131.

Dietterich, T.G. (1967). Ensemble Methods in Machine Learning. *First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science,Springer Verlag, New York* 1-15.

Dagliyan, O., Uney-Yuksektepe, F., Kavakli, I.H., Turkay, A. (2011). Optimization Based Tumor Classification from Microarray Gene Expression Data *PLoS ONE,* 6(2), e14579.

Doksum, K. and Samarov, A. (1995). Nonparametric Estimation of Global Functionals and a Measure of the Explanatory Power of Covariates in Regression. *The Annals of Applied Statistics,* 23, 1443-1473.

Drucker, H. and Cortes, C. (1996). Boosting decision trees. *Advances in Neural Information Processing Systems,* 8, 479-485.

Essawi, M. L., EL-AZIM, S. A., Morsy, A. A., and Hassan, H.A. (2010). Assessment of GSTP1 Gene Methylation in Early Detection of Prostate Cancer in Egyptian Patients. *Med. J. Cairo Univ,* 78(1), 297-301.

Fujii, H., Gabrielson, E., Takagaki, T., Ohtsuji, M., Ohtsuji, N., and Hino, Okio. (2005). Frequent down-regulation of HIVEP2 in human breast cancer. *Breast Cancer Res Treat,* 91(2), 103-12.

Furey, T., Cristianini, N., Duffy, N., Bednarski, D., et al. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics,* 16(10), 906-914.

Friedman, J.,Hastie, T., and Tibshirani, R. (2000). Additive Logistic Regression: a Statistical View of Boosting. *The Annals of Statistics,* 28(2), 337-407.

Garrett, S.C., Varney, K.M., Weber, D.J., and Bresnick, A.R. (2006). S100A4, a mediator of metastasis. *J Biol Chem,*281(2),677-80.

Golub, T. R. Slonim, D.K., and Tamayo, P., et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring *Science,* 286, 531-537.

Guyon, X. (1995). Random Fields on a Network: Modeling, Statistics, and Applications. *Springer-Verlag.*

Ghellal, A., Li, C., Hayes, M., Byrne, G., Bundred, N., Kumar, S. (2000). Prognostic significance of TGF beta 1 and TGF beta 3 in human breast carcinoma. *Anticancer Res,* 20(6B), 4413-8.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res,* 3, 1157-1182.

Hale, T. K., Myers, C., Maitra, R., Kolzau, T., Nishizawa, M., and Braithwaite, A.W. (2000). Maf transcriptionally activates the mouse p53 promoter and causes a p53-dependent cell death. *J Biol Chem,* 275, 17991-17999.

Hanemaaijer, R., Verheijen, J. H., Maguire, T. M., Visser, H., Toet, K., McDermott, E., O'Higgins, N., and Duffy, M. J. (2000). Increased gelatinase-A and gelatinase-B activities in malignant vs. benign breast tumors. *Int. J. Cancer,* 86, 204-207.

Hardle, W. (1994). Applied Nonparametric Regression. *Cambridge.*

Hastie, T., Tibshirani, R., and Friedman, J. (2001). The elements of statistical learning: data mining, inference, and prediction. *Springer-Verlag.*

Hilla, O., Cetinb, Y., Cieslaka A., Magerta, H-J., Forssmann W-G. (1995). A new human guanylate cyclase-activating peptide (GCAP-II, uroguanylin): precursor cDNA and colonic expression. *Biochimica et Biophysica Acta,* 1253, 146-149.

Holland, J. (1975). Adaptation in Natural and Artificial Systems. *University of Michigan Press, Ann Arbor.*

Hooks, S. B., Callihan, P., Altman, M. K., Hurst, J. H., Ali, M. W., and Murph, M. M. (2010). Regulators of G-Protein signaling RGS10 and RGS17 regulate chemoresistance in ovarian cancer cells. *Molecular Cancer,* 9, 289.

Hoshino, Y., Katsuno. Y., Ehata. S., Miyazono, K. (2011). Autocrine TGF- protects breast cancer cells from apoptosis through reduction of BH3-only protein. *J. Biochem,* 149, 55-65.

Hu, Y. L., Zhong, D., Pang, F., Ning, Q. Y., Zhang, Y. Y., Li, G., Wu, J. Z., and Mo, Z. N. (2013). HNF1b is involved in prostate cancer risk via modulating androgenic hormone effects and coordination with other genes. *Genet Mol Res,* 12(2), 1327-35.

Huang, M. and Kecman. V. (2005). Gene extraction for cancer diagnosis by support vector machines - An improvement. *Artificial Intelligence in Medicine,* 35, 185-194.

Itoh, S. and Itoh, F. (2012). Implication of TGF- as a survival factor during tumour development. *J Biochem,*151(6),559-62.

Karakiulakis, G., Papanikolaou, C., Jankovic, S. M., Aletras, A., et al (1997). Increased type IV collagen-degrading activity in metastases originating from primary tumors of the human colon. *Invasion and Metastasis,*173,158-168.

Kim, H. J., Han, J. H., Chang, I. H., Kim, W., and Myung, S.C. (2012). Variants in the HEPSIN gene are associated with susceptibility to prostate cancer. *Prostate Cancer Prostatic Dis,* 15(4), 353-8.

Kittler, J. (1978). Feature set search algorithms. *Pattern Recognition and Signal Processing,* Sijthoff and Noordhoff, Alphen aan den Rijn, Netherlands, 41-60.

Kelly, K. A., Setlur, S. R., Ross, R., Anbazhagan, R., Waterman, P., Rubin, M. A., Weissleder, R. (2008). Detection of Early Prostate Cancer Using a Hepsin-Targeted Imaging Agent. *Cancer Res,* 68, 2286-2291.

Kishino, H. and Waddell, P. (2000). Correspondence analysis of genes and tissue types and finding genetic links from microarray data. *Genome Informatics,*11,83-95.

Khrmann, A., Kammerer, U., Kapp, M., Dietl, J., and Anacker, J. (2009). Expression of matrix metalloproteinases (MMPs) in primary human breast cancer and breast cancer cell lines: New findings and review of the literature. *BMC Cancer,* 9(188).

Lee, J. W., Lee, J. B., Park, M., and Song, S. H. (2005). An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics and Data Analysis,* 48, 869885

Li, F., and Yang, Y. (2005). Analysis of recursive gene selection approaches from microarray data. *Bioinformatics,* 21(19), 3741-3747

Li, S., Harner, E. J., and Adjeroh, D. A. (2011). Random KNN feature selection - a fast and stable alternative to Random Forests. *BMC Bioinformatics,* 12:450

Li, Y. and Agarwal, P. (2009). A Pathway-Based View of Human Diseases and Disease Relationships. *PLoS ONE,* 4(2), e4346

Liu, B., Cui, Q., Tianzi, J., and Ma, S. (2004). A combinational feature selection and ensemble neural network method for classification of gene expression data *BMC Bioinformatics,* 5:136.

Liu, D., Overbey, D., Watkinson, L. D., Daibes-Figueroa, S., and Hoffman, T.J. (2009). In vivo imaging of human colorectal cancer using radiolabeled analogs of the uroguanylin peptide hormone. *Anticancer Res.,* 29, 3777-3783.

Lo, S-H., Chernoff, H., Cong, L., Ding, Y., and Zheng, T. (2008). Discovering Interactions among BRCA1 and Other Candidate Genes Associated with Sporadic Breast Cancer. *Proc. Natl. Acad. Sci. USA,* 105, 12387-12392.

Lo, S-H. and Zheng T. (2002). Backward Haplotype Transmission Association (BHTA) Algorithm a Fast Multiple-Marker Screening Method. *Hum. Hered,* 53, 197-215.

Lo, S-H., and Zheng T. (2004). A Demonstration and Findings of a Statistical Approach through Reanalysis of Inflammatory Bowel Disease Data. *Proc. Natl. Acad. Sci. USA,* 101, 10386-10391.

Lu, T., Pan, Y., Kao, S-Y., Li, C., Kohane, I., Chan, J., and Yankner, BA. (2004). Gene regulation and DNA damage in the ageing human brain. *Nature,* 429, 883-891.

Lu, J., Gosslau, A., Liu, A.Y., and Chen, K.Y. (2008). PCR differential display-based identification of regulator of G protein signaling 10 as the target gene in human colon cancer cells induced by black tea polyphenol theaflavin monogallate. *European Journal Of Pharmacology,* 601, 66-72.

Luo, X., Wang, C. Z., Chen, J., Song, W. X., Luo, J., Tang, N., He, B. C., Kang, Q., Wang, Y., Du, W., He, T. C., and Yuan, C.S. (2008). Characterization of gene expression regulated by American ginseng and ginsenoside Rg3 in human colorectal cancer cells. *Int J Oncol,* 32(5), 975-83.

Mosquera, J. M., Mehra, R., Regan, M. M., Perner, S., et al. (2009). Prevalence of TMPRSS2-ERG fusion prostate cancer among men undergoing prostate biopsy in the United States. *Clin Cancer Res,* 15(14), 4706-4711.

Ng, S. K. A. (2010). To predict disease outcome: clinical risk factors plus genetic-staging for cancer. *Pakistan Journal of Statistics,* 26(1), 171-185.

Oti, M. and Brunner, H. (2007). The modular nature of genetic diseases. *Clin Genet,* 71(1), 1-11.

Phillips, P. C. (2007). Epistasis - the essential role of gene interactions in the structure and evolution of genetic systems. *Nat Rev Genet,* 9(11), 855-867.

Pal, P., Xi, H., Kaushal, R., Sun, G., Jin, C.H., Jin, L., Suarez, B.K., Catalona, W.J., and Deka, R. (2006). Variants in the HEPSIN gene are associated with prostate cancer in men of European origin. *Hum Genet,* 120(2), 187-92.

Porter R. M., Holme T. C., Newman E. L., Hopwood, D., Wilkinson, J. M., and Cuschieri, A. (1993). Monoclonal antibodies to cytoskeletal proteins: an immunohistochemical investigation of human colon cancer. *J. Pathol.,* 170, 435440.

Possemato, R.,Marks, K.M., Shaul, Y.D. et al. (2011). Functional genomics reveal that the serine synthesis pathway is essential in breast cancer. *Nature,* 476, 346-350.

Quinlan, J. R. (1996). Bagging, boosting, and C4.5. *Proceedings of the Thirteenth National Conference on Artificial Intelligence,* 725-730.

Qinyan, Y., Wang, X., Fewell, C., Cameron, J., Zhu, H., et al (2010). MicroRNA miR-155 Inhibits Bone Morphogenetic Protein (BMP) Signaling and BMP-Mediated Epstein-Barr Virus Reactivation?? *J. Virol,* 8413, 6318-6327.

Rao, J. (2002). Targeting actin remodeling profiles for the detection and management of urothelial cancers - A perspective for bladder cancer research. *Front. Biosci.,* 7, e1-8.

Saeys, Y., Inza, I., and Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics,* 23(9), 2507-2517.

Saleem, M., Kweon, M. H., Johnson, J. J., et al. (2006). S100A4 accelerates tumorigenesis and invasion of human prostate cancer through the transcriptional regulation of matrix metalloproteinase 9. *PNAS,* 103(40), 14825-14830.

Scorilas, A., Karameris, A., Arnogiannaki, N., Ardavanis, A., et al (2001). Overexpression of matrix-metalloproteinase-9 in human breast cancer: a potential favourable indicator in node-negative patients. *Brit. J. Cancer,* 84, 1488-1496.

Stute, W. (1984). Asymptotic Normality of Nearest Neighbor Regression Function Estimates. *The Annals of Statistics,* 12, 917926.

Schapire, R. E. (2003). The boosting approach to machine learning: An overview. *Nonlinear Estimation and Classification. Lecture Notes in Statistics Springer, New York ,* 171, 149-171.

Sharad, S., Srivastava, A., Ravulapalli, S., Parker, P., Chen, Y., Li, H., Petrovics, G., and Dobi, A. (2011). Prostate cancer gene expression signature of patients with high body mass index. *Prostate Cancer Prostatic Dis.,* 14(1), 22-9.

Singh, D., Febbo, P.G., Ross, K., et al (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell,* 1, 203-209.

Song, L., Bedo, J., Borgwardt, K.M., et al (2007). Gene selection via the BAHSIC family of algorithms. *Bioinformatics (ISMB),* 23(13), i490-i498.

Sun, H., Yamamoto, M., Mejillano, M., and Yin, H. (1999). Gelsolin, a multifunctional actin regulatory protein. *Biol. Chem ,* 274, 32529-32530.

Tan, A. C., Naiman, D.Q., Xu, L., et al (2005). Simple decision rules for classifying human cancers from gene expression profiles. *Bioinformatics,* 21(20), 3896-3904.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B,* 58, 267-288.

Thompson, V. C., Day, T. K., Bianco-Miotto, T., et al (2012). A gene signature identified using a mouse model of androgen receptor-dependent prostate cancer predicts biochemical relapse in human disease. *Int J Cancer,* 131(3), 662-72.

Tomlins, S. A., Bjartell, A., Chinnaiyan, A. M., et al (2009). ETS gene fusions in prostate cancer: from discovery to daily clinical practice. *Eur Urol,* 56(2), 275-286.

Vainio, P., Gupta, S., Ketola, K. et a. (2011). Arachidonic acid pathway members PLA2G7, HPGD, EPHX2, and CYP4F8 identified as putative novel therapeutic targets in prostate cancer. *Am J Pathol,* 178(2), 525-36.

Van 't Veer, L. J., Dai, H., van de Vijver, M. J., et al(2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature,* 415(6871), 530-6.

Vizoso, F. J., Gonzalez, L. O., Corte, M. D., et al (2007). Study of matrix metalloproteinases and their inhibitors in breast cancer. *Br J Cancer,* 96, 903-911.

Wall, M. E.,Rechtsteiner, A., and Rocha, L. M. (2003). Singular value decomposition and principal component analysis in A Practical Approach to Microarray Data Analysis. *Kluwer: Norwell, MA,* 91-109.

Wang, H., Lo, S-H., Zheng, T., and Hu, I. (2012). Interaction-Based Feature Selection and Classification for High-Dimensional Biological Data. *Bioinformatics,* 28(19), 2407-2411.

Wang, H., Zhang, H., Dai, Z., et al (2013). TSG: a new algorithm for binary and multi-class cancer classification and informative genes selection. *BMC Medical Genomics,* 6(Suppl 1):S3.

Wahde, M., and Szallasi, Z. (2006). A survey of methods for classification of gene expression data using evolutionary algorithms. *Expert Review of Molecular Diagnostics,* 6(1), 101-110.

Wasserman, L., and Roeder, K. (2009). High-Dimension Variable Selection. *The Annals of Statistics,* 37, 2178-2201.

Wei, X. and Li, K-C. (2010). Exploring the within- and between-class correlation distributions for tumor classification. *PNAS,* 107(15), 6737-6742.

Wei, W., Gayatry, M., and Michael J. B. (2012). The FGF18/FGFR4 amplicon: Novel therapeutic biomarkers for ovarian cancer. *Cancer Research,* 72(8), Supplement 1.

Winston, J. S., Asch H. L., Zhang, P. J., et al (2001). Downregulation of gelsolin correlates with the progression to breast carcinoma. *Breast Cancer Res,* 65, 11-21.

Wu, Z. S., Wu, Q., Yang, J. H., et al (2008). Prognostic significance of MMP-9 and TIMP-1 serum and tissue expression in breast cancer. *Int J Cancer,* 122, 2050-2056.

Yang, T., Kecman, V., Cao, L., and Zhang, C. (2010). Combining Support Vector Machines and the t-statistic for Gene Selection in DNA Microarray Data Analysis. *In Lecture Notes in Computer Science,* 55-62.

Yankner, B. A. (2000). A century of cognitive decline. *Nature,* 404, 125.

Zervos, A. S., Gyuris, J., and Brent, R. (1993). Mxi1, a protein that specifically interacts with Max to bind Myc-Max recognition sites. *Cell,* 72(2), 223-32.

Zhang, H., Yu, C-Y., Singer, B., and Xiong, M.(2001). Recursive partitioning for tumor classification with gene expression microarray data. *PNAS,* 98(12), 6730-6735.

Zheng, T., Wang, H., and Lo, S-H.(2006). Backward Genotype-Trait Association (BGTA)-Based Dissection of Complex Traits in Case-Control Designs. *Hum. Hered,* 62, 196-212.

Zhu, J. X, McLachlan, G. J., Ben-Tovim Jones, L., and Wood, I. A.(2008). On selection bias with prediction rules formed from gene expression data. *J. Stat. Plann. Infor,* 138, 374-386.

Zhang, X., Lu, X., Shi, Q., et al (2006). Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data. *BMC Bioinformatics,* 7:197.

Zhang, H., Wang, H., Dai, Z., et al(2012) Improving accuracy for cancer classification with a new algorithm for genes selection. *BMC Bioinformatics,* 13:298.