2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

A New Support Vector Machine for Microarray Classification and Adaptive Gene Selection

Juntao Li, Yingmin Jia, Junping Du and Fashan Yu

Abstract— **This paper presents a new support vector machine for simultaneous gene selection and microarray classification. By introducing the adaptive elastic net penalty which is a convex combination of weighted 1-norm penalty and weighted 2-norm penalty, the proposed support vector machine can encourage an adaptive grouping effect and reduce the shrinkage bias for the large coefficients. According to a reasonable correlation between the two regularization parameters, the optimal coefficient paths are shown to be piecewise linear and the corresponding solving algorithm is developed. Experiments are performed on leukaemia data that verify the research results.**

Index Terms— **Gene selection, grouping effect, microarray classification, solution path, support vector machine (SVM).**

I. Introduction

Development of microarray techniques makes it possible to profile gene expression on a whole genome scale and study associations between gene expression and occurrence or progression of common diseases, such as cancer, HIV and heart disease. A typical microarray dataset has a large number of gene expression values (several thousands or even tens of thousands) and a relatively small number of samples (a few dozen). Therefore, besides predicting the correct class for a given sample, another challenge in microarray classification is to identify the relevant genes which contribute most to the classification.

In recent years, a tremendous amount of efforts have been devoted to microarray classification and gene selection (see, [5], [7], [10], [11], [12], [15], [19], [22], [23], [25] and the reference therein). Although many developed machine learning algorithms achieve similar low classification error rates, most of these methods do not select genes in a satisfactory way. The support vector machine [7], [10] and penalized logistic regression [11], [23] are very successful methods for microarray classification. However, they cannot do gene selection automatically and both use either univariate ranking or recursive feature elimination to reduce the number of genes in the final model. Lasso [17] and the 1-norm SVM [1], [22] have been proposed to perform simultaneous classification and variable selection. Because of the nature of the L_1 norm penalty function, the both methods can reduce the coefficients of irrelevant variables to exactly zero, thus achieving automatic variable selection. However, the 1-norm penalty methods cannot reveal the grouping information in dealing with gene-gene interactions and the number of selected genes is upper bounded by the sample size.

The grouping effect is a natural demand for microarray classification. Biologically speaking, complex diseases, such as cancer, are caused by mutations in gene pathways, instead of individual genes. From the statistical point of view, this can be described as a grouping effect, i.e., generating similar coefficients for highly correlated genes. The group lasso [9], [21] has been developed for selecting the highly correlated and relevant variables in groups. However, how to correctly construct genes clusters in advance and identify important genes within each cluster is still a difficult work. By combining the 1-norm penalty and 2-norm penalty, the elastic net penalized methods [19], [20], [25] can produce a sparse model with good prediction accuracy, while encouraging a grouping effect. Although these methods have been successfully applied to microarray data, there are still several challenges:

- (a) Since the elastic net penalized methods tend to automatically include the whole groups into the model once one gene of them is selected, the redundant noise (the correlated and irrelevant genes) may be included in the fitted model. How to automatically identify important genes within each group is a challenging problem.
- (b) The 1-norm shrinkage would produce biased estimator for the large coefficients. How to properly reduce the shrinkage bias for the large coefficients of significant genes is an interesting problem.
- (c) Two regularization parameters are involved in the elastic net. How to appropriately select the two regularization parameters is an important problem.

This paper is devoted to solving the aforementioned challenges. To this end, we first present the adaptive elastic net penalty, based on which, the adaptive huberized support vector machine (AHSVM) is proposed. Then, the AHSVM is shown to encourage an adaptive grouping effect. After that a reasonable correlation of the two regularization parameters is proposed and the optimal coefficient paths are shown to be piecewise linear Finally, we apply AHSVM to leukaemia classification and achieve promising results.

II. Problem formulation

Assume that the training pairs $\{(x_i, y_i), i = 1, \cdots, n\}$ are independently and identically distributed according to an unknown probability distribution $P(x, y)$. For microarray gene

This work is supported by the NSFC (60374001,60727002,60774003), the MOE (20030006003), the COSTIND (A2120061303) and the National 973 Program (2005CB321902).

J. Li and Y. Jia are with the Seventh Research Division, Beihang University (BUAA), Beijing 100191, China. E-mail: juntaolimail@yahoo.com.cn

J. Du is with the Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, School of Computer Science and Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China.

F. Yu is with the School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo 454000, Henan, China.

expression data, x_i represents the expression levels of p genes of the i-th sample tissue and $y_i \in \{-1,+1\}$ codes its binary response. Our goal is to estimate a linear decision function

$$f(x) = \beta_0 + \beta^T x, \tag{1}$$

and hence build the associated classifier

$$\text{Class}(x) = \text{sign}[f(x)] = \text{sign}[\beta_0 + \beta^T x], \tag{2}$$

for predicting the cancer class of a new sample and identifying the relevant genes.

This is a typical "large p, small n" problem, i.e there are a large number of gene expression values and a relatively small number of samples. There are many ways to fit linear classifier (2), including support vector machines, lasso, boosting and logistic regression. These popular learning machines can be formulated into a generic regularized problem by using $Loss + Penalty$ criterion:

$$\hat{\beta}(\lambda_1,\lambda_2) = \arg\min_{\beta} L(y, f(x)) + \lambda_1 J_1(\beta) + \lambda_2 J_2(\beta), \tag{3}$$

where $\lambda_1, \lambda_2 \geq 0$ are the regularization parameters. The popular loss functions used in machine learning are: hinge loss, squared error loss, exponential loss, negative binomial log-likelihood, huber loss, huberized hinge loss and so on. Let

$$J_2(\beta) = \|\beta\|_2^2 = \sum_{j=1}^{p} \beta_j^2, \qquad J_1(\beta) = \|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|.$$

The popular penalties used in machine learning are:

$$\begin{aligned} &\text{2-norm penalty}: && \lambda_2 > 0 \ \text{ and } \ \lambda_1 = 0 \\ &\text{1-norm penalty}: && \lambda_2 = 0 \ \text{ and } \ \lambda_1 > 0 \\ &\text{Elastic net penalty}: && \lambda_2 > 0 \ \text{ and } \ \lambda_1 > 0 \end{aligned}$$

A variety of learning machines can be constructed by combing the aforementioned losses and penalties: the standard SVM [18] (hinge loss+2-norm penalty), 1-norm SVM [22] (hinge loss+1-norm penalty), the ridge regression (squared error loss+2-norm penalty), lasso [17] (squared error loss+1-norm penalty), huber support vector regression [13] (huber loss+2-norm penalty), the naive elastic net [25] (squared error loss+elastic net penalty) and so on. From the shrinking point of view, all the 2-norm penalized methods can reduce the variance of the estimates and improve the prediction accuracy. However, it can not do automatic gene selection and therefore the additional gene selection methods should be used. Due to its singularity at the origin, 1-norm penalty shrinks some of the coefficients to be exactly zero. Thus the 1-norm penalized methods [1], [22], [17] can do simultaneous gene selection and classification. Although these methods have achieved promising results, they lack the ability to reveal the grouping information in dealing with gene-interactions and the number of the selected genes is upper bounded by the sample size. The elastic net penalty not only retains the benefits of the L_1 norm penalty but also tends to generate similar coefficients for highly correlated variables. However, as shown in Introduction, the elastic net penalized methods still suffer several challenges. This paper is devoted to solving the aforementioned challenges.

In the following, we describe our notation used in the paper. All vectors in this paper will be column vectors unless transposed to a row vector by prime T. For $x \in R^p$, $\|x\|_1$ and $\|x\|_2$ will denote the 1- and 2-norms of x. Let $X = \left(x_{(1)}, x_{(2)}, \cdots, x_{(p)}\right)$ be the model matrix, where $x_{(j)} = (x_{1j}, \cdots, x_{nj})^T$, $j = 1, \cdots, p$ are the predictors.

III. MAIN RESULTS

A. Adaptive huberized support vector machine

Given the set of training pairs (x_i, y_i), the coefficient of the marginal regression could be represented as

$$\tilde{\beta}_j = \frac{\sum_{i=1}^{n} x_{ij} y_i}{\sum_{i=1}^{n} x_{ij}^2}, \tag{4}$$

where $i = 1, 2, \cdots, n$, $j = 1, 2, \cdots, p$. Since the magnitude of $\tilde{\beta}_j$ implies the importance of the corresponding gene in some sense, $|\tilde{\beta}_j|$ can be used to produce a rough gene ranking. Define a weight vector as follows

$$w_j = \begin{cases} |\tilde{\beta}_j|^{-1}, & \text{if } \ |\tilde{\beta}_j| \geq \delta \\ 1/\delta, & \text{otherwise} \end{cases} \tag{5}$$

where $0 < \delta \ll 1$ is a given threshold value. Let

$$\sqrt{W} = diag\{\sqrt{w_1}, \sqrt{w_2}, \cdots, \sqrt{w_p}\},$$
$$W = diag\{w_1, w_2, \cdots, w_p\}.$$

By combining the weighted 1-norm penalty and the weighted 2-norm penalty, we propose the adaptive elastic net penalty

$$\frac{\lambda_2}{2} \|\sqrt{W}\beta\|^2 + \lambda_1 \|W\beta\|_1, \tag{6}$$

where $\|\sqrt{W}\beta\|^2 = \sum_{j=1}^{p} w_j \beta_j^2$, $\|W\beta\|_1 = \sum_{j=1}^{p} w_j |\beta_j|$. Applying the adaptive elastic net penalty to the huberized hinge loss, we propose the following adaptive huberized support vector machine(AHSVM)

$$\min_{\beta_0,\beta} \sum_{i=1}^{n} L_{HH}(y_i f(x_i)) + \frac{\lambda_2}{2} \|\sqrt{W}\beta\|^2 + \lambda_1 \|W\beta\|_1, \tag{7}$$

where λ_2, λ_1 are regularization parameters, $f(x_i)$ is the linear decision function (1), and

$$L_{HH}(y_i f(x_i)) = \begin{cases} 0, & \text{if } y_i f(x_i) > 1, \\ (1 - y_i f(x_i))^2/(2t), & \text{if } 1 - t < y_i f(x_i) \leq 1, \\ 1 - y_i f(x_i) - t/2, & \text{otherwise.} \end{cases}$$

Remark 1: As shown in [13], [19], [20], hinge loss function and huberized hinge loss function have similar shape and hence have similar classification performance. Most importantly, the huberized hinge loss function is differentiable everywhere, which is not the case for hinge loss function. This differentiability can significantly reduce the computational cost for developing regularization path algorithm, especially for the initial setup.

Remark 2: The weighted 1-norm penalty is used to adaptively penalize each component such that the coefficients

of irrelevant genes are shrunken to zero, while reducing the shrinkage bias for the large coefficients of significant variables(see, [24]). The rationale behind the weighted 2-norm penalty is to adaptively penalize the coefficients of significant genes such that the highly correlated genes are adaptively selected in groups according to their ranking significance(see, III-B).

Remark 3: It should be noted that there are several popular methods [7], [23] for ranking genes in terms of their classification performance. They will work more efficiently than the marginal regression method in the sense of gene ranking. However, introducing marginal regressor in AHSVM is not to select genes but to adaptively penalize coefficients of gene expressions, and the real genes selection is automatically achieved by 1-norm shrinkage.

Since the huberized hinge loss function has different definitions in different regions, we define each region as

- $\mathscr{R} = \{: y_i f(x_i) > 1\}$ ($\mathscr{R}$ for right of the Elbow),
- $\mathscr{E} = \{i : 1-t \le y_i f(x_i) \le 1\}$ ($\mathscr{E}$ for Elbow),
- $\mathscr{L} = \{i : y_i f(x_i) \le 1-t\}$ ($\mathscr{L}$ for left of the Elbow),

define the indices for non-zero β_j as the active set $\mathscr{A}$ as

- $\mathscr{A} = \{j : \beta_j \neq 0,\ j = 1,2,\cdots,p\}$ ($\mathscr{A}$ for active set).

B. Adaptive grouping effect

Theorem 1: Let $\hat{\beta}_0$ and $\hat{\beta}$ denote the optimal solution for (7). Let $x_{(j)}$ and $x_{(l)}$ be the gene expressions corresponding to $\tilde{\beta}_j$ and $\tilde{\beta}_l$. If $|\tilde{\beta}_j| \ge \delta$, $|\tilde{\beta}_l| \ge \delta$ and $\hat{\beta}_j \hat{\beta}_l > 0$, then we have

$$|\hat{\beta}_j - \hat{\beta}_l| \le \frac{1}{\lambda_2} \sum_{i=1}^{n} \left| |\tilde{\beta}_j| x_{ij} - |\tilde{\beta}_l| x_{il} \right|. \tag{8}$$

Furthermore, if $x_{(j)}$ and $x_{(l)}$ are centered and normalized, then we have

$$|\hat{\beta}_j - \hat{\beta}_l| \le \frac{\sqrt{n}}{\lambda_2} \sqrt{1-\gamma\rho} \sqrt{\tilde{\beta}_j^2 + \tilde{\beta}_l^2} \tag{9}$$

where $\rho = x_{(j)}^T x_{(l)} = \sum_{i=1}^n x_{ij} x_{il}$, and $\gamma = 2|\tilde{\beta}_j \tilde{\beta}_l| / (\tilde{\beta}_j^2 + \tilde{\beta}_l^2)$.

Proof: If $\hat{\beta}_j \hat{\beta}_l > 0$, then $\hat{\beta}_j$ and $\hat{\beta}_l$ are non-zero and $sign(\hat{\beta}_j) = sign(\hat{\beta}_l)$. Let

$$L(\lambda_1, \lambda_2, \beta) = L_{HH}(y_i f(x_i)) + \frac{\lambda_2}{2} \|\sqrt{W}\beta\|^2 + \lambda_1 \|W\beta\|_1. \tag{10}$$

Since problem (7) is an unconstrained convex optimization problem, the derivatives of objective function with respect to $\hat{\beta}$ satisfie

$$\left. \frac{\partial L(\lambda_1, \lambda_2, \beta)}{\partial \beta_k} \right|_{\beta = \hat{\beta}, \beta_0 = \hat{\beta}_0} = 0 \quad \text{if} \quad \hat{\beta}_k \neq 0. \tag{11}$$

Hence, for $\tilde{\beta}_j \ge \delta$ and $\hat{\beta}_j \neq 0$, we have

$$\sum_{i \in \mathscr{E}} \frac{1}{t} (f(x_i) - y_i) x_{ij} - \sum_{i \in \mathscr{L}} y_i x_{ij} + \lambda_2 w_j \hat{\beta}_j + \lambda_1 w_j sign(\hat{\beta}_j) = 0. \tag{12}$$

Since $w_j = |\tilde{\beta}_j|^{-1} > 0$, (12) is equivalent to

$$\hat{\beta}_j = \frac{1}{\lambda_2} \left[\sum_{i \in \mathscr{E}} \frac{1}{t} (y_i - f(x_i)) |\tilde{\beta}_j| x_{ij} + \sum_{i \in \mathscr{L}} y_i |\tilde{\beta}_j| x_{ij} - \lambda_1 sign(\hat{\beta}_j) \right]. \tag{13}$$

Analogously, for $\tilde{\beta}_l \ge \delta$ and $\hat{\beta}_l \neq 0$, we have

$$\hat{\beta}_l = \frac{1}{\lambda_2} \left[\sum_{i \in \mathscr{E}} \frac{1}{t} (y_i - f(x_i)) |\tilde{\beta}_l| x_{il} + \sum_{i \in \mathscr{L}} y_i |\tilde{\beta}_l| x_{il} - \lambda_1 sign(\hat{\beta}_l) \right]. \tag{14}$$

Note that $sign(\hat{\beta}_j) = sign(\hat{\beta}_l)$. Subtracting (14) from (13) gives

$$\hat{\beta}_j - \hat{\beta}_l = \frac{1}{\lambda_2} \left[\sum_{i \in \mathscr{E}} \frac{1}{t} (y_i - f(x_i)) (|\tilde{\beta}_j| x_{ij} - |\tilde{\beta}_l| x_{il}) + \sum_{i \in \mathscr{L}} y_i (|\tilde{\beta}_j| x_{ij} - |\tilde{\beta}_l| x_{il}) \right]. \tag{15}$$

Since $y_i f(x_i) > 1 - t > 0$ for $i \in \mathscr{E}$, we have $sign(y_i) = sign(f(x_i))$. On the other hand, from $y_i f(x_i) < 1$ for $i \in \mathscr{E}$ and $|y_i| = 1$, we have $1 - t < |y_i f(x_i)| = |f(x_i)| \le 1$. Hence, it can be easily obtain that

$$|y_i - f(x_i)| = |y_i| - |f(x_i)| < 1 - (1-t) = t \tag{16}$$

From (15) and (16), we have

$$\begin{aligned} &|\hat{\beta}_j - \hat{\beta}_l| \\ &\le \frac{1}{\lambda_2} \left[\sum_{i \in \mathscr{E}} \frac{1}{t} |(y_i - f(x_i))| \left| |\tilde{\beta}_j| x_{ij} - |\tilde{\beta}_l| x_{il} \right| \right. \\ &\left. + \sum_{i \in \mathscr{L}} |y_i| \left| |\tilde{\beta}_j| x_{ij} - |\tilde{\beta}_l| x_{il} \right| \right] \\ &\le \frac{1}{\lambda_2} \left[\sum_{i \in \mathscr{E}} \left| |\tilde{\beta}_j| x_{ij} - |\tilde{\beta}_l| x_{il} \right| + \sum_{i \in \mathscr{L}} \left| |\tilde{\beta}_j| x_{ij} - |\tilde{\beta}_l| x_{il} \right| \right] \\ &\le \frac{1}{\lambda_2} \sum_{i=1}^{n} \left| |\tilde{\beta}_j| x_{ij} - |\tilde{\beta}_l| x_{il} \right| = \frac{1}{\lambda_2} \left\| |\tilde{\beta}_j| x_{(j)}^T - |\tilde{\beta}_l| x_{(l)}^T \right\|_1. \end{aligned} \tag{17}$$

Furthermore, if $x_{(j)}$ and $x_{(l)}$ are centered and normalized, it can be easily obtained that

$$\begin{aligned} \left\| |\tilde{\beta}_j| x_{(j)}^T - |\tilde{\beta}_l| x_{(l)}^T \right\|_1 &\le \sqrt{n} \left\| |\tilde{\beta}_j| x_{(j)}^T - |\tilde{\beta}_l| x_{(l)}^T \right\|_2 \\ &\le \sqrt{n} \sqrt{\tilde{\beta}_j^2 + \tilde{\beta}_l^2 - 2\tilde{\beta}_j \tilde{\beta}_l x_{(j)}^T x_{(l)}} \\ &= \sqrt{n} \sqrt{\tilde{\beta}_j^2 + \tilde{\beta}_l^2} \sqrt{1-\gamma\rho}. \end{aligned} \tag{18}$$

From (17) and (18), (9) can be easily obtained. This completes the proof. ■

It should be noted that *Theorem 1* still holds if $|\tilde{\beta}_j| \ge \delta$ and $|\tilde{\beta}_l| < \delta$. The only difference is substituting $|\delta|$ for $\tilde{\beta}_l$. If $|\tilde{\beta}_j| < \delta$ and $|\tilde{\beta}_l| < \delta$, we have the following *Corollary*:

Corollary 1: Let $\hat{\beta}_0$ and $\hat{\beta}$ denote the optimal solution for (7). Let $x_{(j)}$ and $x_{(l)}$ be the gene expressions corresponding to $\tilde{\beta}_j$ and $\tilde{\beta}_l$. If $|\tilde{\beta}_j| \le \delta$, $|\tilde{\beta}_l| \le \delta$ and $\hat{\beta}_j \hat{\beta}_l > 0$, then we have

$$|\hat{\beta}_j - \hat{\beta}_l| \le \frac{\delta}{\lambda_2} \left\| \bar{x}_j - \bar{x}_l \right\|_1 = \frac{\delta}{\lambda_2} \sum_{i=1}^{n} \left| x_{ij} - x_{il} \right|. \tag{19}$$

Furthermore, if $x_{(j)}$ and $x_{(l)}$ are centered and normalized, then we have

$$|\hat{\beta}_j - \hat{\beta}_l| \le \frac{\delta\sqrt{n}}{\lambda_2} \sqrt{2(1-\rho)}. \tag{20}$$

According to the results in [19], [20], given the same assumption as *Theorem 1* or *Corollary 1*, the doubly regularized support vector machine (DRSVM) and the hybrid huberized support vector machine (HHSVM) satisfy

$$|\hat{\beta}_j - \hat{\beta}_l| \le \frac{\sqrt{n}}{\lambda_2} \sqrt{2(1-\rho)}. \tag{21}$$

This implies that the both support vector machines tend to generate similar coefficients for highly correlated genes, i.e., encouraging a grouping effect. However, since the whole groups will be automatically included into the model once one of them is selected, the fitted model may include more redundant genes. From Theorem 1 and Corollary 1, the proposed AHSVM can adaptively control the grouping effect according to sample correlation and the ranking significance of genes. For the case $|\tilde{\beta}_i|, |\tilde{\beta}_j| < \delta \ll 1$, from (20) and (21), we know that AHSVM has stronger grouping effect compared with HHSVM and DRSVM. This means that the more genes (the bigger size of group) are removed together by 1-norm shrinkage if they are less important to the classification. According to the mean inequality, $\gamma = 1$ if and only if $|\tilde{\beta}_i| = |\tilde{\beta}_j|$. From Theorem 1, AHSVM can assign identical coefficients to the genes only if the sample correlation $\rho = 1$ and the ranking significance $|\tilde{\beta}_i| = |\tilde{\beta}_j| > \delta$. It is easy to see that the more genes with similar ranking significance ($|\tilde{\beta}_i| \approx |\tilde{\beta}_j|$), the bigger size of the selected groups. This means that AHSVM can adaptively select the highly correlated genes by evaluating their ranking significance. This also implies that adaptive gene selection can be automatically achieved within the selected group.

C. The solution path

Although SVM training algorithms [2], [8], [14], [16] have been widely studied, most of the methods cannot deal the model selection problem. Recently, a novel approach has emerged that seeks to explore the entire solution path for all parameter values without having to re-train the model multiple times [3], [4], [11], [19], [20], [22], [24]. Unfortunately, these methods could not be efficiently extended to the AHSVM since two regularization parameters are involved.

In general, increasing λ_1 tends to eliminate more irrelevant variables, and increasing λ_2 makes the grouping effect more prominent. It seems that λ_1 and λ_2 are not correlative. Note that eliminating more variables should encourage more stronger grouping effect. Hence, one natural correlation is that λ_2 should decrease with decreasing λ_1. Motivated by the aforementioned idea, we see λ_2 is the monotonically nondecreasing constant function of λ_1. Similar to [19], if we continuously decrease λ_1, some of sets of $\mathscr{L}$, $\mathscr{E}$, $\mathscr{R}$, and $\mathscr{A}$ will change. We call this an event, and four types of events may occur:

- 1. A point reaches the boundary between $\mathscr{L}$ and $\mathscr{E}$;
- 2. A point reaches the boundary between $\mathscr{R}$ and $\mathscr{E}$;
- 3. A parameter β_j becomes zero, i.e., j leaves $\mathscr{A}$;
- 4. A zero-valued parameter β_j becomes non-zero.

We use the superscript l to index the sets above immediately after lth event has occurred. Suppose $|\mathscr{A}^l| = m$, and let β_0^l, β^l, λ_1^l, λ_2^l be the values of these parameters at the point of entry. Likewise f^l is the function at this point. We continuously decreases λ_1 until it reaches 0. For $\lambda_1^l \ge \lambda_1 > \lambda_1^{l+1}$, we let

$$\lambda_2^l = \max\left\{ a,\ b - \frac{b-a}{\ln(e+\lambda_1^l)} \right\} \tag{22}$$

where $0 < a < b$ are the given constants. It should be noted that λ_2 is the monotonically nondecreasing constant function of λ_1 in the interval $[\lambda_1^{l+1}, \lambda_1^l]$.

Let $\bar{x}_{\mathscr{A}\mathscr{E}1}$ and $\bar{x}_{\mathscr{A}\mathscr{E}2}$ be the $m \times 1$ vectors with jth entrys $\sum_{i\in\mathscr{E}} x_{ij}$ and $\sum_{i\in\mathscr{E}} x_{ij} sign(\beta_j^l)/w_j$ for $j \in \mathscr{A}$, respectively. let A be the $m \times m$ matrix with jkth entry

$$A_{jk} = \begin{cases} \sum_{i\in\mathscr{E}} x_{ij} x_{ik} sign(\beta_j^l)/(t w_j), & \text{for } j \ne k \\ \left(\sum_{i\in\mathscr{E}} \frac{x_{ij}^2}{t w_j} + \lambda_2 \right) sign(\beta_j^l), & \text{for } j = k \end{cases}$$

for $j,k \in \mathscr{A}$.

Theorem 2: If the regularization parameters of AHSVM satisfy (22), then the optimal coefficient $\hat{\beta}_0(\lambda_1)$ and $\hat{\beta}(\lambda_1)$ of (7) are piecewise linear with respective to regularization parameter λ_1. Furthermore, for $\lambda_1^l \ge \lambda_1 > \lambda_1^{l+1}$, we have

$$\begin{cases} \hat{\beta}_0 = \hat{\beta}_0^l + (\lambda_1 - \lambda_1^l)\bar{b}_0 \\ \hat{\beta}_j = \hat{\beta}_j^l + (\lambda_1 - \lambda_1^l)\bar{b}_j, \quad \text{for} \quad j \in \mathscr{A}^l \end{cases} \tag{23}$$

$$f(x_i) = f^l(x_i) + (\bar{b}_0 + \sum_{j\in\mathscr{A}} x_{ij}\bar{b}_j)(\lambda_1 - \lambda_1^l) \tag{24}$$

where $\bar{b}_j$ is the $j+1$ element of vector $\bar{A}_l^{-1} 1^\alpha$, and $\bar{A}_l$ and 1^α are defined as

$$\bar{A}_l = \begin{pmatrix} m & \bar{x}_{\mathscr{A}\mathscr{E}1}^T \\ \frac{1}{t}\bar{x}_{\mathscr{A}\mathscr{E}2} & A \end{pmatrix}, \qquad 1^\alpha = \begin{pmatrix} 0 \\ 1_m \end{pmatrix}.$$

Proof: Since (7) is an unconstrained convex optimization problem, the derivatives of objective function with respect to $\hat{\beta}_0$, $\hat{\beta}$ satisfy

$$\begin{aligned} \left. \frac{\partial L(\lambda_1,\lambda_2,\beta)}{\partial \beta_0} \right|_{\beta=\hat{\beta},\beta_0=\hat{\beta}_0} &= 0 \\ \left. \frac{\partial L(\lambda_1,\lambda_2,\beta)}{\partial \beta_k} \right|_{\beta=\hat{\beta},\beta_0=\hat{\beta}_0} &= 0 \quad \text{if} \quad \hat{\beta}_k \ne 0. \end{aligned} \tag{25}$$

Note that λ_2 is constant value and sets $\mathscr{E}^l$, $\mathscr{R}^l$, $\mathscr{L}^l$, $\mathscr{A}^l$ will not change for $\lambda_1^l \geq \lambda_1 > \lambda_1^{l+1}$. Hence, we have

$$\sum_{i\in\mathscr{E}^l} \frac{1}{t}(\hat{\beta}_0 + \sum_{k\in\mathscr{A}^l} x_{ik}\hat{\beta}_k - y_i) - \sum_{i\in\mathscr{L}^l} y_i = 0 \tag{26}$$

$$\sum_{i\in\mathscr{E}^l} \frac{1}{t}(\hat{\beta}_0 + \sum_{k\in\mathscr{A}^l} x_{ik}\hat{\beta}_k - y_i)x_{ij} - \sum_{i\in\mathscr{L}^l} y_i x_{ij} + \lambda_2^l w_j \hat{\beta}_j + \lambda_1 w_j sign(\hat{\beta}_j) = 0. \tag{27}$$

for $j \in \mathscr{A}^l$. For $\lambda_1 = \lambda_1^l$, we also have

$$\sum_{i\in\mathscr{E}^l} \frac{1}{t}(\hat{\beta}_0^l + \sum_{k\in\mathscr{A}^l} x_{ik}\hat{\beta}_k^l - y_i) - \sum_{i\in\mathscr{L}^l} y_i = 0 \tag{28}$$

$$\sum_{i\in\mathscr{E}^l} \frac{1}{t}(\hat{\beta}_0^l + \sum_{k\in\mathscr{A}^l} x_{ik}\hat{\beta}_k^l - y_i)x_{ij} - \sum_{i\in\mathscr{L}^l} y_i x_{ij} + \lambda_2^l w_j \hat{\beta}_j^l + \lambda_1^l w_j sign(\hat{\beta}_j^l) = 0. \tag{29}$$

for $j \in \mathscr{A}^l$. Subtracting equation (28) from (26) gives

$$\sum_{i\in\mathscr{E}^l} (\hat{\beta}_0 - \hat{\beta}_0^l + \sum_{k\in\mathscr{A}^l} x_{ik}(\hat{\beta}_k - \hat{\beta}_k^l)) = 0. \tag{30}$$

Note that $sign(\hat{\beta}_j) = sign(\hat{\beta}_j^l)$ for $j \in \mathscr{A}^l$ and $\lambda_1^l \geq \lambda_1 > \lambda_1^{l+1}$ (otherwise, $\hat{\beta}_j$ will becomes zero and therefore the set A has changed). Subtracting equation (29) from (27) gives

$$\sum_{i\in\mathscr{E}^l} \frac{1}{t}(\hat{\beta}_0 - \hat{\beta}_0^l + \sum_{k\in\mathscr{A}^l} x_{ik}(\hat{\beta}_k - \hat{\beta}_k^l))x_{ij} + \lambda_2^l w_j(\hat{\beta}_j - \hat{\beta}_j^l) + (\lambda_1 - \lambda_1^l) w_j sign(\hat{\beta}_j^l) = 0. \tag{31}$$

for $j \in \mathscr{A}^l$. Let $\hat{\tilde{\beta}}$ and $\hat{\tilde{\beta}}^l$ be the vectors which elements are $\hat{\beta}_k, \hat{\beta}_k^l$ for $k \in \mathscr{A}^l$, respectively. Note that

$$\sum_{i\in\mathscr{E}^l}\sum_{k\in\mathscr{A}^l} x_{ik}(\hat{\beta}_k - \hat{\beta}_k^l)) = \sum_{k\in\mathscr{A}^l}\sum_{i\in\mathscr{E}^l} x_{ik}(\hat{\beta}_k - \hat{\beta}_k^l)) = \bar{x}_{\mathscr{A}\mathscr{E}1}^T(\hat{\tilde{\beta}} - \hat{\tilde{\beta}}^l).$$

Hence, (30) can be rewritten as

$$m(\hat{\beta}_0 - \hat{\beta}_0^l) + \bar{x}_{\mathscr{A}\mathscr{E}1}^T(\hat{\tilde{\beta}} - \hat{\tilde{\beta}}^l) = 0 \tag{32}$$

Analogously, (31) for $j \in \mathscr{A}^l$ can be rewritten as

$$\frac{1}{t}\bar{x}_{\mathscr{A}\mathscr{E}2}(\hat{\beta}_0 - \hat{\beta}_0^l) + A(\hat{\tilde{\beta}} - \hat{\tilde{\beta}}^l) = (\lambda_1 - \lambda_1^l)1_m \tag{33}$$

if $\bar{A}_l$ has full rank, (23) can be easily obtained by solving linear system of equations (32) and (33). Furthermore, substituting (32) and (33) into (1) gives (24). This completes the proof. ■

Similar to [4], [19], [20], [22], our algorithm starts from $\lambda_1 \to \infty$; continuously decreases λ_1; solves the optimal piecewise linear solution along this path. The main algorithm that computes the whole solution path $\hat{\beta}_0, \hat{\beta}$ proceeds as follows:

1) : Calculate $\hat{\beta}_0^0, \hat{\beta}^0$, λ_1^0, λ_2^0, $\mathscr{E}^0$, $\mathscr{L}^0$, $\mathscr{R}^0$, $\mathscr{A}^0$.

2) : Find the λ_1^{l+1} and λ_2^{l+1}.
   - Let $\lambda_2^l = \max\left\{a,\ b - \frac{b-a}{\ln(e+\lambda_1^l)}\right\}$.
   - Calculate $\hat{\beta}_0$, $\hat{\beta}_j$ and $f(x_i)$ for $i = 1,2,\cdots,n$, $j = 1,2,\mathscr{A}^l$ according to (23) and (24).
   - Determine the step size d_1 for the first event.
   - Determine the step size d_2 for the second event.
   - Determine the step size d_3 for the third event.
   - Determine the step size d_4 for the fourth event.
   - Determine the step size d for event which happens first. $d = \min\{d_1, d_2, d_3, d_4\}$.

3) If any one of the following termination criterion is met, then stop the algorithm.
   - The generalized correlation reduces to zero.
   - Two classes have been perfectly separated.
   - A pre-specified maximum iteration number is reached.

4) Otherwise, let $l = l+1$, $\lambda_1^{l+1} = \lambda_1^l - d$, $\lambda_2^{l+1} = \max\left\{a,\ b - \frac{b-a}{\ln(e+\lambda_1^{l+1})}\right\}$ and update $\hat{\beta}_0^l, \hat{\beta}^l$, $\mathscr{E}^l$, $\mathscr{L}^l$, $\mathscr{R}^l$. Then goto the step 2.

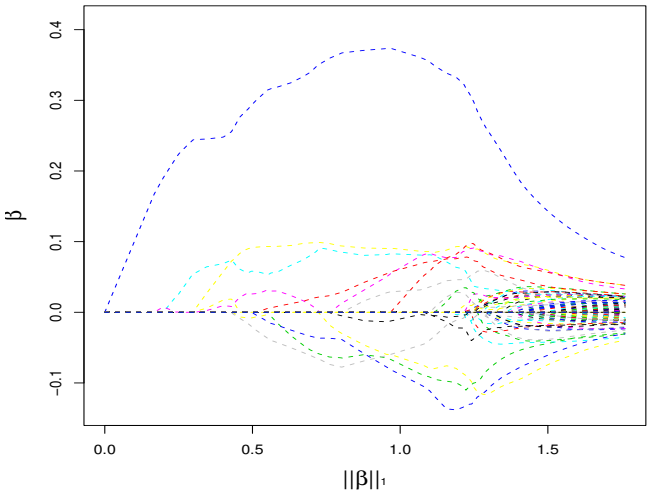

Fig. 1. the coefficients paths of AHSVM.

It should be noted that the solving procedure is similar to [19] if the natural correlation between λ_1 and λ_2 is satisfied, the main difference (and also the difficulty) is to calculate the step size d for event which happens first. We skip the detailed calculation for the limits of space.

IV. Experiments on Leukaemia Data

To illustrate the effectiveness of AHSVM for microarray, we perform experiments on the classic leukaemia data (Golub et al., 1999). This dataset consists of 38 training data and 34 test data for two types of acute leukemia, acute myeloid leukaemia (AML) and acute lymphoblastic leukaemia (ALL). Each datum is a vector of $p = 7129$ genes. The goal is to construct a diagnostic rule to predict the type of leukaemia based on the expression level of those 7129 genes. The original data and experimental methods are available at $http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi$.

To make the computation more manageable, we use the same pre-processing of the Golub et al (1999). Each time that

Table 1: The top 10 genes selected by AHSVM

Estimate	Gene ID	Gene description
0.07546321	D50915_at	KIAA0125 gene
-0.05854534	M11722_at	Terminal transferase mRNA
0.07034328	M19507_at	MPO Myeloperoxidase
0.20476500	M27891_at	CST3 Cystatin C (amyloid angiopathy and cerebral hemorrhage)
-0.04414178	U05259_rnal_at	MB-1 gene
-0.09935767	Z14982_rnal_at	MHC-encoded proteasome subunit gene LAMP7-E1 gene (proteasome subunit LMP7) extracted from H.sapiens gene for major histocompatibility complex encoded proteasome subunit LMP7
0.06957877	X95735_at	Zyxin
-0.08471228	U89922_s_at	LTB Lymphotoxin-beta
0.04175153	M63438_s_at	GLUL Glutamate-ammonia ligase (glutamine synthase)
0.04874105	U01317_cds4_at	Delta-globin gene extracted from Human beta globin region on chromosome 11

Table 2: Summary of the leukaemia classification results

Method	Tenfold CV error	Test error	Number of genes
Golub	3/38	4/34	50
SVM	2/38	1/34	31
HHSVM	0/38	0/34	84
AHSVM	0/38	0/34	46

a model is fitted, we first select the 3571 most "significant" genes as the predictors. Then, we compute the regularization solution path according to the algorithm in III-C, where parameters b, a, t are selected as $2, 0.05$, and 0.95, respectively. Fig.1 shows the coefficient paths by using $\|\beta\|_1$ as the parameter. The optimal model is given when $\|\beta\|_1$ is equal to 1.376056. The corresponding regularization parameter λ_1 is 0.0373583 and the number of the selected genes is 46.

Table1 lists the top 10 genes that have been selected by AHSVM. Gene M19507_at is highly correlated with gene X95735_at (positive correlation), and they have the similar ranking significance. Hence, their corresponding estimates are almost equal. Analogously, gene U05259_rnal_at and gene M63438_s_at have highly negative correlation and sharing the similar ranking significance. Hence, their corresponding estimates have the similar absolute values with the different sign. Compared with the optimal HHSVM which need 84 genes in [19], AHSVM selects less genes. Table 2 compares AHSVM with several competitors including Golubs method, SVM and HHSVM. AHSVM gives the best classification.

V. Conclusion

The adaptive huberized support vector machine for simultaneous microarray classification and gene selection has been proposed in this paper. It is shown that AHSVM not only can reduce the shrinkage bias for the large coefficients, but also can control the size of the selected groups and therefore automatically identify important genes within each group. Furthermore, based on a reasonable correlation of the two regularization parameters, the optimal coefficients are proved to be piecewise linear with the single regularization parameter and an efficient solution path algorithm is developed. We compare AHSVM with other methods on leukemia dataset and AHSVM achieves promising results on both classification and gene selection.

References

[1] P. Bradley, O. Mangasarian, Feature selection via concave minimization and support vector machines. *Proceedings of the 15th International Conference on Machine Learning*, 1998.

[2] J. Dong, A. Krzyzak, and C. Y. Suen, Fast SVM training algorithm with decomposition on very large data sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp 603-618, 2005.

[3] B. Efron, T. Hastie, I. Johnston, and R. Tibshirani, Least angle regression , *Annals of Statistics*, vol. 32, pp 407-451, 2004.

[4] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu, The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, vol. 5, pp 1391-1415, 2004.

[5] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing and M. Caligiuri, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, *Science*, vol, 286, pp 531-536, 1999.

[6] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, vol, 3, pp 1157-1182, 2003.

[7] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, Gene selection for cancer classification using support vector machines, *Machine Learning*, vol, 46, pp 389-422, 2002.

[8] L. Jiao, L. Bo, and L. Wang, Fast sparse approximation for least square support vector machines. *IEEE Transactions on Neural Networks*, vol. 18, pp 685-697, 2007.

[9] S. Ma, X. Song, J. Huang, Supervised group lasso with applications to microarray data analysis. *Bioinformatics*, vol, 8, 2007.

[10] S. Mukherjee, R. Tamayo, D. Slonim, A. Verri, T. Golub, J. Mesirov, and T. Poggio, Support vector machine classification of microarray data (Technical Report). Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 2000.

[11] M. Park, T. Hastie, Penalized logistic regression for detecting gene interactions. *Biostatistics*, vol, 9, pp 30-50, 2008.

[12] M. Park, T. Hastie, R. Tibshirani, Averaged gene expressions for regression. *Biostatistics*, vol, 8, pp 212-227, 2007.

[13] S. Rosset, and J. Zhu, Piecewise linear regularized solution paths. *Annals of Statistics*, vol, 35, pp 1012-1030, 2007.

[14] B. Schokpof, A. Smola, R. C. Williamson, and P. L. Bartlett, New support vector algorithms. *Neural Computation*, vol. 12, pp 1207-1245, 2000.

[15] M. Segal, K. Dahlquist and B. Conklin, Regression approach for microarray data analysis. *Journal of Computational Biology*, vol, 10, pp 961-980, 2003.

[16] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya and K. R. K. Murthy, Improvements to the SMO algorithm for SVM regression. *IEEE Transactions on Neural Networks*, vol. 11, pp 1188-1194, 2000.

[17] R. Tibshirani, Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society, Series B*, vol, 58, pp 267-288, 1996.

[18] V. Vapnik, *The nature of statistical learning theory*. New York: Springer-Verlag, 1995.

[19] L. Wang, J. Zhu and H. Zou, Hybrid huberized support vector machines for microarray classification and gene selection. *Bioinformatics*, vol, 24, pp 412-419, 2008.

[20] L. Wang, J. Zhu, and H. Zou. The doubly regularized support vector machine. *Statistica Sinica*, vol, 16, pp 589-615, 2006.

[21] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, vol, 68, pp 49-67, 2006.

[22] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, 1-norm support vector machines. *Advances in Neural Information Processing Systems*, vol, 16, pp 49-56, 2004.

[23] J. Zhu, T. Hastie, Classification of gene microarrays by penalized logistic regression. *Biostatistics*, vol, 46, pp 505-510, 2004.

[24] H. Zou, An Improved 1-norm Support Vector Machine for Simultaneous Classification and Variable Selection. *Eleventh International Conference on Artificial Intelligence and Statistics*, 2007.

[25] H. Zou, and T. Hastie, Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society, Series B*, vol, 67, pp 301-320, 2005.