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, \ldots, n\} are independently and identically distributed according to an unknown probability distribution $P(x, y)$. For microarray gene
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
$$f(x) = \beta_0 + \beta^T x,$$
(1)
and hence build the associated classifier
$$\text{Class}(x) = \text{sign}[f(x)] = \text{sign}[\beta_0 + \beta^T x],$$
(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:
$$\hat{\beta}(\lambda_1, \lambda_2) = \arg\min_{\beta} L(y, f(x)) + \lambda_1 J_1(\beta) + \lambda_2 J_2(\beta),$$
(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, and relatively binomial log-likelihood, huber loss, huberized hinge loss and so on. Let
$$J_2(\beta) = \|\beta\|_2^2 = \sum_{j=1}^{p} \beta_j^2, \quad J_1(\beta) = \|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|.$$
The popular penalties used in machine learning are:

2-norm penalty: $\lambda_2 > 0$ and $\lambda_1 = 0$
1-norm penalty: $\lambda_2 = 0$ and $\lambda_1 > 0$
Elastic net penalty: $\lambda_2 > 0$ and $\lambda_1 > 0$

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), 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 L1 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 $x \in \mathbb{R}^n$, $\|x\|_1$ and $\|x\|_2$ will denote the 1- and 2-norms of $x$. Let $X = (x_{(1)}, x_{(2)}, \cdots, x_{(p)})$ be the model matrix, where $x_{(j)} = (x_{ij}, \cdots, x_{nj})^T, j = 1, \cdots, p$ 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
$$\hat{\beta}_j = \frac{\sum_{i=1}^{n} x_{ij} y_i}{\sum_{i=1}^{n} x_{ij}^2},$$
(4)
where $i = 1, 2, \cdots, n, j = 1, 2, \cdots, p$. Since the magnitude of $\hat{\beta}_j$ implies the importance of the corresponding gene in some sense, $|\hat{\beta}_j|$ can be used to produce a rough gene ranking. Define a weight vector as follows
$$w_j = \begin{cases} |\hat{\beta}_j|^{-1}, & \text{if } |\hat{\beta}_j| \geq \delta \\ 1/\delta, & \text{otherwise} \end{cases}$$
(5)
where $0 < \delta \ll 1$ is a given threshold value. Let
$$\sqrt{W} = \text{diag}\{\sqrt{w_1}, \sqrt{w_2}, \cdots, \sqrt{w_p}\}, \quad W = \text{diag}\{w_1, w_2, \cdots, w_p\}.$$
By combining the weighted 1-norm penalty and the weighted 2-norm penalty, we propose the adaptive elastic net penalty
$$\frac{\lambda_2}{2} \|\sqrt{W}\beta\|^2 + \lambda_1 \|W\beta\|_1,$$
(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 (AH SVM)
$$\min_{\beta_0, \beta} \sum_{i=1}^{n} L_{HH}(y_i, f(x_i)) + \frac{\lambda_2}{2} \|\sqrt{W}\beta\|^2 + \lambda_1 \|W\beta\|_1,$$
(7)
where $\lambda_2, \lambda_1$ are regularization parameters, $f(x_i)$ is the linear decision function (1), and
$$L_{HH}(y_i, f(x_i)) = \begin{cases} 0, & \text{if } y_i f(x_i) > 1 \\ (1 - y_i f(x_i))^2/(2\sigma^2), & \text{if } 1 - t < y_i f(x_i) \leq 1 \\ 1 - y_i f(x_i) - t/2, & \text{otherwise} \end{cases}$$

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 shrunk 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:

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

define the indices for non-zero \( \beta_j \) as the active set \( \mathcal{A} \) as:

\[ \mathcal{A} = \{ j : \beta_j \neq 0, \; j = 1, 2, \ldots, p \} \quad (\mathcal{A} \text{ for active set}). \]

B. Adaptive grouping effect

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

\[ |\hat{\beta}_j - \hat{\beta}_l| \leq \frac{1}{\lambda_2} \sum_{i=1}^{n} |\hat{\beta}_j x_{ij} - |\hat{\beta}_l| x_{il}|. \]  

(8)

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

\[ |\hat{\beta}_j - \hat{\beta}_l| \leq \sqrt{n} \frac{\lambda_2}{\lambda_2} \sqrt{1 - \gamma \rho} \sqrt{\hat{\beta}_j^2 + \hat{\beta}_l^2} \]

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

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

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

(10)

Since problem (7) is an unconstrained convex optimization problem, the derivatives of objective function with respect to \( \beta \) satisfy

\[ \frac{\partial L(\lambda_1, \lambda_2, \beta)}{\partial \beta_k} \bigg|_{\beta = \beta, \beta_k = 0} = 0 \quad \text{if} \quad \hat{\beta}_k \neq 0. \]  

(11)

Hence, for \( \beta_j > \delta \) and \( \beta_l \neq 0 \), we have

\[ \sum_{i \in \mathcal{E}} \frac{1}{t} (f(x_i) - y_i) x_{ij} - \sum_{i \in \mathcal{L}} y_i x_{ij} + \lambda_2 w_j \hat{\beta}_j + \lambda_1 w_j \text{sign}(\hat{\beta}_j) = 0. \]  

(12)

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

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

(13)

Analogously, for \( \hat{\beta}_l \geq \delta \) and \( \beta_l \neq 0 \), we have

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

(14)

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

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

(15)

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

\[ |y_i - f(x_i)| = |y_i| - |f(x_i)| < 1 - (1 - t) = t \]  

(16)

From (15) and (16), we have

\[ |\hat{\beta}_j - \hat{\beta}_l| \leq \frac{1}{\lambda_2} \left[ \sum_{i \in \mathcal{E}} \frac{1}{t} ||y_i - f(x_i)|| |\hat{\beta}_j| x_{ij} - |\hat{\beta}_l| x_{il}| + \sum_{i \in \mathcal{L}} y_i |\hat{\beta}_j| x_{ij} - |\hat{\beta}_l| x_{il} \right] \]

\[ \leq \frac{1}{\lambda_2} \left[ \sum_{i \in \mathcal{E}} (|\hat{\beta}_j| x_{ij} - |\hat{\beta}_l| x_{il}) + \sum_{i \in \mathcal{L}} (|\hat{\beta}_j| x_{ij} - |\hat{\beta}_l| x_{il}) \right] \]

\[ \leq \frac{1}{\lambda_2} \sum_{i=1}^{n} (|\hat{\beta}_j| x_{ij} - |\hat{\beta}_l| x_{il}) = \frac{1}{\lambda_2} \left[ ||\hat{\beta}_j x_{(j)}^T - |\hat{\beta}_l| x_{(l)}^T||_1 \right]. \]  

(17)

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

\[ ||\hat{\beta}_j x_{(j)}^T - |\hat{\beta}_l| x_{(l)}^T||_1 \leq \sqrt{n} \left( ||\hat{\beta}_j x_{(j)}^T - |\hat{\beta}_l| x_{(l)}^T||_2 \right) \]

\[ \leq \sqrt{n} \left( \hat{\beta}_j^2 + \hat{\beta}_l^2 - 2 \hat{\beta}_j \hat{\beta}_l x_{(j)}^T x_{(l)} \right) \]

\[ = \sqrt{n} \left( \hat{\beta}_j^2 + \hat{\beta}_l^2 \right) \sqrt{1 - \gamma \rho}. \]  

(18)

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

It should be noted that Theorem 1 still holds if \( \beta_j > \delta \) and \( \beta_l < \delta \). The only difference is substituting \( |\delta| \) for \( |\beta_l| \) and \( |\beta_j| < \delta \), 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 \( \hat{\beta}_j \) and \( \hat{\beta}_l \). If \( |\hat{\beta}_j| \leq \delta \), \( |\hat{\beta}_l| \leq \delta \) and \( \hat{\beta}_j \hat{\beta}_l > 0 \), then we have
\[
|\hat{\beta}_j - \hat{\beta}_l| \leq \frac{\delta}{\lambda_2} \left| \bar{x}_j - \bar{x}_l \right|_{1} = \frac{\delta}{\lambda_2} \sum_{j=1}^{n} |x_{ij} - x_{il}|.
\] (19)
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 \( |\hat{\beta}_j|, |\hat{\beta}_l| < \delta \ll 1 \), 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 important to the classification. According to the mean inequality, \( \gamma = 1 \) if and only if \( |\hat{\beta}_j| = |\hat{\beta}_l| \). From Theorem 1, AHSVM can assign identical coefficients to the genes only if the sample correlation \( \rho = 1 \) and the ranking significance \( |\hat{\beta}_j| = |\hat{\beta}_l| > \delta \). It is easy to see that the more genes with similar ranking significance (\( |\hat{\beta}_j| \approx |\hat{\beta}_l| \)), 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 \( \lambda_1 \) tends to eliminate more irrelevant variables, and increasing \( \lambda_2 \) makes the grouping effect more prominent. It seems that \( \lambda_1 \) and \( \lambda_2 \) are not correlated. Note that eliminating more variables should encourage more stronger grouping effect. Hence, one natural correlation is that \( \lambda_2 \) should decrease with decreasing \( \lambda_1 \). Motivated by the aforementioned idea, we see \( \lambda_2 \) is the monotonically nondecreasing constant function of \( \lambda_1 \). Similar to [19], if we continuously decrease \( \lambda_1 \), some of sets of \( L, \delta, \mathcal{R}, \) and \( \mathcal{A} \) will change. We call this an event, and four types of events may occur:

- 1. A point reaches the boundary between \( L \) and \( \delta \);
- 2. A point reaches the boundary between \( \mathcal{R} \) and \( \delta \);
- 3. A parameter \( \beta_j \) becomes zero, i.e., \( j \) leaves \( \mathcal{A} \);
- 4. A zero-valued parameter \( \beta_j \) becomes non-zero.

We use the superscript \( l \) to index the sets above immediately after \( l \)th event has occurred. Suppose \( |\mathcal{A}^l| = m \) and let \( \hat{\beta}^0_1, \hat{\beta}^1, \lambda^0_1, \lambda^1_1 \) be the values of these parameters at the point of entry. Likewise \( f^{l} \) is the function at this point. We continuously decreases \( \lambda_1 \) until it reaches 0. For \( \lambda^0_1 \geq \lambda_1 > \lambda^{l+1}_1 \), we let
\[
\lambda^1_1 = \max \left\{ a, b - \frac{b - a}{\ln(e + \frac{1}{\lambda^0_1})} \right\}
\] (22)
where \( 0 < a < b \) are the given constants. It should be noted that \( \lambda^1_1 \) is the monotonically nondecreasing constant function of \( \lambda_1 \) in the interval \( [\lambda^{l+1}_1, \lambda^0_1] \).

Let \( \bar{x}_{j,j} \) and \( \bar{x}_{j,j+1} \) be the \( m \times 1 \) vectors with \( j \)th entries \( \sum_{i \in \mathcal{E}} x_{ij} \) and \( \sum_{i \in \mathcal{E}} x_{ij} \text{sign}(\hat{\beta}^1_i)/w_{ij} \) for \( j \in \mathcal{A} \), respectively. Let \( A \) be the \( m \times m \) matrix with \( j \)th entry
\[
A_{jk} = \left\{ \sum_{i \in \mathcal{E}} x_{ij} x_{ik} \text{sign}(\hat{\beta}^1_i)/(tw_{ij}), \text{ for } j \neq k \right\}
\]
for \( j, k \in \mathcal{A} \).

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 respect to regularization parameter \( \lambda_1 \). Furthermore, for \( \lambda^l_1 \geq \lambda_1 > \lambda^{l+1}_1 \), we have
\[
\left\{ \begin{array}{l}
\hat{\beta}_0 = \hat{\beta}^0_1 + (\lambda_1 - \lambda^0_1)\bar{b}_0 \\
\hat{\beta}_j = \hat{\beta}^1_j + (\lambda_1 - \lambda^0_1)\bar{b}_j, \text{ for } j \in \mathcal{A} \\
f(x_i) = f^l(x_i) + (\bar{b}_0 + \sum_{j \in \mathcal{A}} x_j \bar{b}_j)(\lambda_1 - \lambda^0_1) \\
\end{array} \right.
\] (23)
where \( \bar{b}_j \) is the \( j + 1 \) element of vector \( \bar{\lambda}^{-1}_1 \mathbf{1}^\alpha \), and \( \bar{\lambda} \) and \( 1^\alpha \) are defined as
\[
\bar{\lambda} = \left( \frac{m}{\frac{1}{\lambda^l_1} \bar{x}_{\mathcal{A}^l,\mathcal{A}^l} \bar{x}_{\mathcal{A}^l,\mathcal{A}^l}} A \right), \quad \alpha = \left( \begin{array}{c}
0 \\
1_m \\
\end{array} \right).
\]

Proof: Since (7) is an unconstrained convex optimization problem, the derivatives of objective function with respect to \( \hat{\beta}_0, \hat{\beta}_j \) satisfy
\[
\frac{\partial L(\lambda_1, \lambda_2, \beta)}{\partial \hat{\beta}_j} = 0 \quad \text{if } \hat{\beta}_j \neq 0.
\] (25)
Note that $\lambda_2$ is constant value and sets $\delta^l$, $\beta^l$, $L^l$, $A^l$ will not change for $\lambda^l \geq \lambda_1 > \lambda^{l+1}_1$. Hence, we have

$$\sum_{i \in \mathcal{E}^l} \frac{1}{l}(\hat{\beta}_0 + \sum_{k \in \mathcal{E}^l} x_k \hat{\beta}_k - y_i) - \sum_{i \in \mathcal{Y}^l} y_i = 0$$ (26)

$$\sum_{i \in \mathcal{E}^l} \frac{1}{l}(\hat{\beta}_0 + \sum_{k \in \mathcal{E}^l} x_k \hat{\beta}_k - y_i)x_{ij} - \sum_{i \in \mathcal{Y}^l} y_ix_{ij}$$
$$+ \lambda^l_2 w_j \hat{\beta}_j + \lambda^l_1 w_j \text{sign}(\hat{\beta}_j) = 0.$$ (27)

for $j \in \mathcal{E}^l$. Subtracting equation (28) from (27) gives

$$\sum_{i \in \mathcal{E}^l} \frac{1}{l}(\hat{\beta}_0 - \hat{\beta}_0^l + \sum_{k \in \mathcal{E}^l} x_k (\hat{\beta}_k - \hat{\beta}_k^l)) - \sum_{i \in \mathcal{Y}^l} y_i = 0$$ (28)

$$\sum_{i \in \mathcal{E}^l} \frac{1}{l}(\hat{\beta}_0^l + \sum_{k \in \mathcal{E}^l} x_k \hat{\beta}_k^l - y_i)x_{ij} - \sum_{i \in \mathcal{Y}^l} y_ix_{ij}$$
$$+ \lambda^l_2 w_j \hat{\beta}_j^l + \lambda^l_1 w_j \text{sign}(\hat{\beta}_j^l) = 0.$$ (29)

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

$$\sum_{i \in \mathcal{E}^l} (\hat{\beta}_0 - \hat{\beta}_0^l + \sum_{k \in \mathcal{E}^l} x_k (\hat{\beta}_k - \hat{\beta}_k^l)) = 0.$$ (30)

Note that $\text{sign}(\hat{\beta}_j) = \text{sign}(\hat{\beta}_j^l)$ for $j \in \mathcal{E}^l$ and $\lambda^l_1 \geq \lambda_1 > \lambda^{l+1}_1$ (otherwise, $\hat{\beta}_j$ will become zero and therefore the set A has changed). Subtracting equation (29) from (27) gives

$$\sum_{i \in \mathcal{E}^l} \frac{1}{l}(\hat{\beta}_0 - \hat{\beta}_0^l + \sum_{k \in \mathcal{E}^l} x_k (\hat{\beta}_k - \hat{\beta}_k^l))x_{ij} + \lambda^{l+1}_2 w_j (\hat{\beta}_j - \hat{\beta}_j^l)$$
$$+ (\lambda_1 - \lambda^l_1) w_j \text{sign}(\hat{\beta}_j^l) = 0.$$ (31)

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

$$\sum_{i \in \mathcal{E}^l} x_k (\hat{\beta}_k - \hat{\beta}_k^l)) = \sum_{i \in \mathcal{E}^l} \sum_{k \in \mathcal{E}^l} x_k (\hat{\beta}_k - \hat{\beta}_k^l) = \mathbf{x}^T_{\mathcal{E}^l} (\hat{\beta} - \hat{\beta}^l).$$

Hence, (30) can be rewritten as

$$m(\hat{\beta}_0 - \hat{\beta}_0^l) + \mathbf{x}^T_{\mathcal{E}^l} (\hat{\beta} - \hat{\beta}^l) = 0.$$ (32)

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

$$\frac{1}{l} \mathbf{x}^T_{\mathcal{E}^l} (\hat{\beta}_0 - \hat{\beta}_0^l) + A (\hat{\beta} - \hat{\beta}^l) = (\lambda_1 - \lambda^l_1) 1_m.$$ (33)

If $\tilde{A}_l$ 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 $\lambda_l \to \infty$; continuously decreases $\lambda_1$; solves the optimal piecewise linear solution along this path. The main algorithm that computes the whole solution path $\hat{\beta}_0, \hat{\beta}_j^l$ proceeds as follows:

1) Calculate $\hat{\beta}_0^0, \beta^0_0, \lambda^0_1, \lambda^0_2, \delta^0, \beta^0, \mathcal{E}^0, \mathcal{Y}^0, \mathcal{A}^0$. Then goto step 2.

2) Find the $\lambda^{l+1}_1$ and $\lambda^{l+1}_2$.
   - Let $\lambda^{l+1}_2 = \max \left\{ a, \frac{b-a}{\ln(c+\lambda^{l+1}_1)} \right\}$.
   - Calculate $\hat{\beta}_0, \hat{\beta}_j^l$ and $f(x_i)$ for $i = 1, 2, \ldots, n$, $j = 1, 2, \mathcal{A}^l$ according to (23) and (24).

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 $l = l + 1$, $\lambda^{l+1}_1 = \lambda^l_1 - d$, $\lambda^{l+1}_2 = \max \left\{ a, \frac{b-a}{\ln(c+\lambda^l_1)} \right\}$ and update $\hat{\beta}_0^l, \beta^l, \delta^l, L^l, A^l$. Then goto step 2.

It should be noted that the solving procedure is similar to [19] if the natural correlation between $\lambda_1$ and $\lambda_2$ 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

![Fig. 1. the coefficients paths of AHSVM.](image-url)
Table 1: The top 10 genes selected by AHSVM

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Gene ID</th>
<th>Gene description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.07546321</td>
<td>D50915</td>
<td>Terminal transferase mRNA</td>
</tr>
<tr>
<td>0.07034228</td>
<td>M19507</td>
<td>MPO Myeloperoxidase</td>
</tr>
<tr>
<td>0.20476500</td>
<td>M27891</td>
<td>CST3 Cystatin C (anoyl antigen and cerebral hemorrhage)</td>
</tr>
<tr>
<td>-0.04414775</td>
<td>U05259</td>
<td>MB-1 gene</td>
</tr>
<tr>
<td>-0.09935767</td>
<td>Z14982</td>
<td>MHC-encoded proteasome subunit gene LAMP7-E1 gene (proteasome subunit LMP7) extracted from H.sapiens gene for major histocompatibility complex encoded proteasome subunit LMP7</td>
</tr>
<tr>
<td>0.00957875</td>
<td>X95735</td>
<td>Zyxin</td>
</tr>
<tr>
<td>-0.08471228</td>
<td>U89922</td>
<td>LTB Lymphotoxin-beta</td>
</tr>
<tr>
<td>0.04175153</td>
<td>M63438</td>
<td>GLUL Glutamate-ammonia ligase (glutamine synthase)</td>
</tr>
<tr>
<td>0.04874105</td>
<td>U01317</td>
<td>Delta-globin gene extracted from Human beta globin region on chromosome 11</td>
</tr>
</tbody>
</table>

Table 2: Summary of the leukaemia classification results

<table>
<thead>
<tr>
<th>Method</th>
<th>Tenfold CV error</th>
<th>Test error</th>
<th>Number of genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Golub</td>
<td>3/38</td>
<td>4/34</td>
<td>50</td>
</tr>
<tr>
<td>SVM</td>
<td>2/38</td>
<td>1/34</td>
<td>31</td>
</tr>
<tr>
<td>HHSVM</td>
<td>0/38</td>
<td>0/34</td>
<td>84</td>
</tr>
<tr>
<td>AHSVM</td>
<td>0/38</td>
<td>0/34</td>
<td>46</td>
</tr>
</tbody>
</table>

data is developed. We compare AHSVM with other methods on leukemia dataset and AHSVM achieves promising results on both classification and gene selection.

**REFERENCES**


