A Graph-based Elastic Net for Variable Selection and Module Identification for Genomic Data Analysis

Zheng Xia and Xiaobo Zhou
Department of Radiology,
The Methodist Hospital Research Institute,
Houston, TX 77030, USA
Weill Cornell Medical College,
New York, NY 10065, USA
Email: zxia@tmhs.org and xzhou@tmhs.org

Wei Chen and Chunqi Chang
Department of Electrical and Electronic Engineering,
The University of Hong Kong,
Hong Kong, PR China
Email: chenwei@eee.hku.hk and cqchang@eee.hku.hk

***Abstract*—Recently a network-constraint regression model[1] is proposed to incorporate the prior biological knowledge to perform regression and variable selection. In their method, a l_1-norm of the coefficients is defined to impose sparse, meanwhile a Laplacian operation on the biological graph is designed to encourage smoothness of the coefficients along the network. However the grouping effect of their Laplacian smoothness operation only exits when the two connected genes both have positive or negative effects on the response. To overcome this problem, we proposed to apply the Laplacian operation on the absolute values of the coefficients to take account of the positive and negative effects. Here, we call the presented method as graph-based elastic net (GENet) because the proposed method has similar grouping effect with elastic net(ENet)[2] except the smoothness of two coefficients are specified by the network in GENet. Further, an efficient algorithm which has same spirit with LARS [3] is developed to solve our optimization problem. Simulation studies showed that the proposed method has better performance than network-constrained regularization without absolute values. Application to Alzheimer's disease(AD) microarray gene-expression dataset identified several subnetworks on Kyoto Encyclopedia of Genes and Genomes(KEGG) transcriptional pathways that are related to progression of AD. Many of those findings are confirmed by published literatures.**

***Keywords*-Laplacian graph; Elastic Net; pathway;**

I. Introduction

As more and more biological data such as microarray and SNParray available, linking high-dimensional genomic data with biological processes and diseases to build a prediction model for interpretation and diagnosis are becoming the central problem in genomic research. Generally the problem can be formulated as a linear regression model with responses vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^T$ and p predictors $\mathbf{x}_j = (x_{1j}, \ldots, x_{nj})^T, j = 1, \ldots, p$. The response can be binary to represent the two conditions such as 'disease' and 'control' or quantitative to indicate the progress of the disease. We consider the classic linear regression model where the response $\boldsymbol{y}$ is predicted by

$$\hat{\boldsymbol{y}} = \hat{\beta}_0 + \mathbf{x}_1\hat{\beta}_1 + \ldots + \mathbf{x}_p\hat{\beta}_p \quad (1)$$

The vector of coefficients $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)$ can be estimated by a model fitting procedure. In genomic data analysis, $n \ll p$ always holds. To address the curse of dimensionality, many new regularized methods have been developed to identify the predictors that are involved in the biological process, such as lasso [4] and elastic net [2]. Among these methods, the elastic net and the fused-lasso are particularly suitable for analysis of genomic data which both accounted the biological fact that genes involve the same pathway have more similar contributions to the response and should be selected/discarded together with higher probability.

One limitation of all these popular approaches is that those methods are developed purely from computational or algorithmic points without delivering any prior biological knowledge or information. Some well-known pathway databases include KEGG, Reactome (www.reactome.org). These pathways are often interconnected and form a network, which can be represented as graphs, where the vertices of the graphs are genes or gene products and the edges of the graphs indicate some regulatory relationship between the genes. Several statistical methods have been developed to utilize the pathways or network information[5]. Further, [1] proposed a network-constrained regularization procedure for fitting linear-regression models and for variable selection where the regularization is a combination of the lasso penalty and a penalty induced by Laplace matrix of the graph. Such a procedure can select subnetworks of correlated features along the network connection instead of the whole pairwise correlation of elastic net. However, the grouping effect of their Laplacian smoothness operation does not work in the condition, where one of the two connected genes has positive effect and the other one has negative effect on the response. Here we utilize the Laplacian operation on the absolute values of the coefficients which can cope with this problem appropriately. Our proposed procedure is a generalization of elastic net (ENet) to address cases with the prior network available. The biological network is used to constraint the difference between the absolute values of coefficients of two

connected genes.

The network constraint on the absolute values of the coefficients renders the optimization solute difficult because absolute operation is not differentiable. At the time of this writing, we notice [6] proposed a novel constraint term which also render $|\hat{\boldsymbol{\beta}}_i|$ and $|\hat{\boldsymbol{\beta}}_j|$ of the two linked nodes similar. However, they used boosted lasso [7] to solve their optimization problem which can not get the exact whole entire regularization path. We can also turn our problem into a quadratic programming by decomposing the coefficients into the summation of the positive and negative parts. The shortcoming of the decomposition is there will be $2 * p$ parameters to be estimated while the small sample number is not incread, which is problematic in $n \ll p$ case. Here, we developed an efficient algorithm to solve the GENet problem which follows the same spirit of homotopy method LARS [3] to give the entire regularization path.

II. Methods

Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_p]$ be the predictor matrix. Without loss of the generality we assume the predictors are standardized and the response is centered. Consider a network that is represented by a weighted graph $G = (V, E, W)$, where V is the set of vertices that correspond to the p predictors, $E = \{u \sim v\}$ is the set of edges indicating that the predictors u and v are linked on the network and there is an edge between u and v and W is the weights of the edges, where $w(u, v)$ denotes the weight of edge $e = (u \sim v)$. Define the degree of the vertex v as $d_v = \sum_{u \sim v} w(u, v)$ where $\sum_{u \sim v}$ denotes the sum over all connected pairs on the network. We say u is an isolated vertex if $d_u = 0$. We define the normalized Laplacian matrix L for G with the uvth element defined by

$$L(u,v) = \begin{cases} 1 - w(u,v)/d_u & \text{if } u = v \text{ and } d_u \neq 0, \\ -w(u,v)/\sqrt{d_u d_v} & \text{if } u \text{ and } v \text{ are adjacent,} \\ 0 & \text{otherwise.} \end{cases}$$

The matrix L is always non-negative definite and its corresponding set of the eigenvalues or spectrum reflects many properties of the graph [8].

For fixed λ_1 and λ_2, [1] defined their network-constrained regularization criterion.

$$\begin{aligned} & L_{nk}(\lambda_1, \lambda_2, \boldsymbol{\beta}) \\ & \quad = (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^T (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda_1 ||\boldsymbol{\beta}||_1 \\ & \qquad + \lambda_2 \sum_{u \sim v} \left(\frac{\beta_u}{\sqrt{d_u}} - \frac{\beta_v}{\sqrt{d_v}}\right)^2 w(u,v) \end{aligned} \tag{2}$$

where $||\boldsymbol{\beta}||_1 = \sum_{j=1}^{p} |\boldsymbol{\beta}_j|$ is the l_1-norm, which induces a sparse solution [4], and the third term induces a smooth solution of $\boldsymbol{\beta}$ on the network. The biological motivation of this regularization is the assumption that genes that are linked on the networks to have similar functions and therefore smoothed. However,the effects of two connected genes on response y will have similar amplitudes but different directions (positive or negative). [1] did not take account of the positive and negative effects. To overcome this shortcoming, we propose the following GENet model with regularization on the absolute value of the coefficients.

$$\begin{aligned} & L(\lambda_1, \lambda_2, \boldsymbol{\beta}) \\ & \quad = (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^T (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda_1 \sum_{j=1}^{p} |\beta_j| \\ & \qquad + \lambda_2 \sum_{u \sim v} \left(\frac{|\beta_u|}{\sqrt{d_u}} - \frac{|\beta_v|}{\sqrt{d_v}}\right)^2 w(u,v) \\ & \quad = (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^T (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda_1 ||\boldsymbol{\beta}||_1 + \lambda_2 |\boldsymbol{\beta}|^T L |\boldsymbol{\beta}| \end{aligned} \tag{3}$$

where $|\boldsymbol{\beta}| = (|\beta_1|, \ldots, |\beta_p|)^T$. The absolute value network-constrained regularized estimator $\hat{\boldsymbol{\beta}}$ is defined as the minimizer of Equation (3)

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \{L(\lambda_1, \lambda_2, \boldsymbol{\beta})\} \tag{4}$$

Let $\lambda = \lambda_1 + \lambda_2$ and $\alpha = \lambda_2/\lambda$, the last two regularization terms can be written as

$$\lambda P_\alpha(\boldsymbol{\beta}) = \lambda \left((1-\alpha)||\boldsymbol{\beta}||_1 + \alpha \sum_{u \sim v} \left(\frac{|\beta_u|}{\sqrt{d_u}} - \frac{|\beta_v|}{\sqrt{d_v}}\right)^2 w(u,v) \right)$$

We call $P_\alpha(\boldsymbol{\beta})$ the absolute value network-constrained regularization, in which the second term imposes smoothness of the absolute values of coefficients $\boldsymbol{\beta}$ over the network. The coefficients β is re-scaled in order to account for different degrees of the vertices on the network, allowing the genes with more connections (e.g. the hub genes) to have larger coefficients so that small changes of expressions of such genes can lead to large changes in the response. Our regularization term constrains the coefficients of the connected genes have similar amplitude effects to the response $\boldsymbol{y}$ but allowing different directions: positive or negative. The weight $w(u, v)$ can be binary or quantitative value to cope with the weighted interaction network such as STRING . In this paper, we just discuss the network with binary weight of edges $w(u, v) \in \{0, 1\}$.

Figure 1 shows contours for four penalty functions for a bivariate argument $\boldsymbol{\beta} = (\beta_1, \beta_2)$,where $\alpha = 0.1$ for the elastic net,network-constraint and our absolute value network-constraint penalties. From Fig. 1, we can see that network-constrain penalty has no group effect in the second and fourth quadrant where the shapes are very similar with the lasso. The elastic net and our proposed method have the group effect on the all four quadrants and our method has larger group effect than elastic net with the same α.

III. Optimization algorithm

Inspired by the LARS [3] and its extended general piecewise linear solution strategy [9] ,we develop an efficient algorithm to solve the absolute value network-constrained

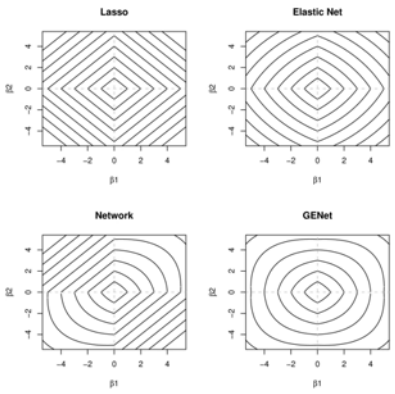

Figure 1. Contours for four penalty functions for a bivariate argument $\boldsymbol{\beta} = (\beta_1, \beta_2)$. The upper left shows contours of the lasso penalty. The upper right shows contours of the elastic net penalty. The lower left shows the contours of the network-constrainted penalty and the lower right shows the contours of our proposed term, both for $\alpha = 0.1$.

regularization problem on the whole solute path for every possible value of λ_1. There are two parameters λ_1 and λ_2 to be tuned. In this section,we propose an efficient algorithm,which solves the entire solute path for every possible vale of λ_1 with λ_2 fixed. The algorithm is based on the fact that the solution $\hat{\boldsymbol{\beta}}$ is a piecewise linear function with respect with λ_1.

Let's first define the active set $\mathcal{A} = \{j : \beta_j \neq 0, j = 1, 2, \ldots, p\}$. The solution of problem (3) satisfies the following equations.

$$\begin{aligned}\frac{\partial L}{\partial \beta_j} &= \sum_i (\mathbf{x}_{i\cdot}^T \boldsymbol{\beta} - y_i)\mathbf{x}_{ij} \\ &+ \lambda_2 \text{sgn}(\beta_j) L_{j\cdot} |\boldsymbol{\beta}| + \lambda_1 \text{sgn}(\beta_j) = 0, \text{for } j \in \mathcal{A}\end{aligned} \tag{5}$$

where $\mathbf{x}_{i\cdot}$ is the ith row of the prediction matrix $\mathbf{X}$, $L_{j\cdot}$ indicates the jth row of Laplacian matrix L and $|\boldsymbol{\beta}|$ is a vector with each component $|\boldsymbol{\beta}|_j = |\beta_j|$. When set $\mathcal{A}$ is fixed, the solution $\beta_j, j \in \mathcal{A}$ are linear functions of λ_1. However the set $\mathcal{A}$ will eventually change as λ_1 decreases. The structure of linear system (6) will also change to become another system. The algorithm decrease λ_1 from ∞ to 0 and solve the solutions along this path. When $\lambda_1 = \infty$ all the $\beta_j = 0$ and $\mathcal{A} = \emptyset$. Reducing λ_1 implies the magnitude of $\boldsymbol{\beta}$ will increase. Keeping reducing λ_1, a critical point λ_1^0 will occur where exactly one $\beta_j (j = 1, \ldots, p)$ will become non-zero to join $\mathcal{A}$. Since Equation (6) must hold for any $j \in \mathcal{A}$, the critical point λ_1^0 is determined by:$\lambda_1^0 = \max_{j \in \{1,\ldots,p\}} |y_i \mathbf{x}_{ij}|$. The first element of $\mathcal{A}$ can be identified as: $\hat{j} = \arg\max_{j \in \{1,\ldots,p\}} |y_i \mathbf{x}_{ij}|$ and the sign for $\beta_{\hat{j}}$ is : $\text{sgn}(\beta_{\hat{j}}) = -\text{sgn}(y_i \mathbf{x}_{i\hat{j}})$. We use k to represent the iteration number. Now the state in the initial ($k = 0$) stage is $\mathcal{A}^0 = \{\hat{j}\}$,$\lambda_1^k = \lambda_1^0$ and $\beta_j = 0 (j = 1, \ldots, p)$.

A. Solution path

Now we have the initial state. The algorithm continuously decreases λ_1 until it reaches 0 and give the solutions along this path. Let $\lambda_1 = \lambda_1^k + \Delta\lambda_1$, where $\Delta\lambda_1 < 0$. When λ_1 is reduced by a small enough amount, the active set $\mathcal{A}$ will not change due to the linear change of $\boldsymbol{\beta}$ with respect to λ_1 in a small region. Therefore, based on the system, the derivative of $\beta_j (j \in \mathcal{A})$ with respect to λ_1 can be solved from the following equations:

$$\begin{aligned}&\sum_i (\sum_{l \in \mathcal{A}} x_{il} \frac{\Delta \boldsymbol{\beta}_k}{\Delta \lambda_1} x_{ij}) + \lambda_2 \text{sgn}(\beta_j) \sum_{l \in \mathcal{A}} L_{jl} \text{sgn}(\beta_l) \frac{\Delta \beta_k}{\Delta \lambda_1} \\ &+ \text{sgn}(\beta_j) = 0, \text{ for } j \in \mathcal{A}\end{aligned} \tag{6}$$

The $|\mathcal{A}|$ unknown $\frac{\Delta \beta_j}{\Delta \lambda_1} (j \in \mathcal{A})$ can be uniquely determined by the above $|\mathcal{A}|$ linear equations as long as the system is non-singular. Then in this linear change region, the solutions are linear in λ_1

$$\beta_j = \beta_j^k + \frac{\Delta \beta_j}{\Delta \lambda_1} (\lambda_1 - \lambda_1^k), \text{ for } j \in \mathcal{A}$$

If λ_1 is kept reducing, the active set $\mathcal{A}$ will change. Here the change refers to the following two *events*.

- Event A: a non-zero coefficient β_j leaves $\mathcal{A}$ (becomes zero)
- Event B: a zero-valued coefficient β_j joins $\mathcal{A}$ (becomes non-zero)

The event A will appear when a non-zero β_j approaches to 0. The step size to become zero for each $\beta_j (j \in \mathcal{A})$ can be calculated by: $\Delta\lambda_1^j = -\beta_j^k / \frac{\Delta \beta_j}{\Delta \lambda_1}$. So the step size for the event A to occur is:

$$\Delta\lambda_{1,\text{A}} = \max\{\Delta\lambda_1^j : j \in \mathcal{A}, \Delta\lambda_1^j \leq 0\}$$

To determine the step size of event B. Let's first make some transformation on the Equation.

$$\begin{aligned}&\sum_i (\mathbf{x}_i^T \boldsymbol{\beta} - y_i) x_{ij} + \lambda_2 L_{jj} \beta_j \\ &= -(\lambda_2 \sum_{l \neq j} L_{jl} |\beta_i| + \lambda_1) \text{sgn}(\beta_j), \text{ for } j \in \mathcal{A}\end{aligned} \tag{7}$$

Taking some parts from the above equation, we define

$$\begin{aligned}C_j &= \sum_i (\mathbf{x}_i^T \boldsymbol{\beta} - y_i) x_{ij} + \lambda_2 L_{jj} \beta_j \\ D_j &= \lambda_2 \sum_{l \neq j} L_{jl} |\beta_l| + \lambda_1\end{aligned}$$

where $j = 1, \ldots, p$. From Equation (7), we can infer that:

- $\text{sgn}(\beta_j) = -\text{sgn}(C_j)\text{sgn}(D_j)$, for $j \in \mathcal{A}$
- $|C_j| = |D_j|$, for $j \in \mathcal{A}$ and $|C_j| \neq |D_j|$ for $j \in \mathcal{A}^c$

Notice that when $\Delta\lambda_1$ is sufficiently small, C_j and D_j are also linear function of λ_1:

$$C_j = C_j^k + \left[\sum_i (\sum_{l\in\mathcal{A}} x_{il}\frac{\Delta\beta_l}{\Delta\lambda_1})x_{ij} + \lambda_2 L_{jj}\frac{\Delta\beta_j}{\Delta\lambda_1}\right](\lambda_1 - \lambda_1^k)$$

$$D_j = D_j^k + \left[\lambda_2 \sum_{l\in\mathcal{A}, l\neq j} \text{sgn}(\beta_l)L_{jl}\frac{\Delta\beta_l}{\Delta\lambda_1} + 1\right](\lambda_1 - \lambda_1^k)$$

As λ_1 decreases, the value for a $|C_j|(j \in \mathcal{A}^c)$ will first equal $|D_j|$ and then the corresponding β_j will become non-zero if we further reduce λ_1.

To determine the step size for the event B, we have to calculate the step size for each $j \in \mathcal{A}^c$. Because $|C_j| = |D_j|$ we have $C_j = +D_j$ or $C_j = -D_j$. From $C_j = +D_j$ we can derive

$$\Delta\lambda_{1,+}^j = \frac{C_j^k - D_j^k}{1 + \lambda_2 \sum_{l\in\mathcal{A}} \text{sgn}(\beta_l)L_{jl}\frac{\Delta\beta_l}{\Delta\lambda_1} - \sum_i (\sum_{l\in\mathcal{A}} x_{il}\frac{\Delta\beta_l}{\Delta\lambda_1})x_{ij}}$$

Similarly the following equation is obtained according to $C_j = -D_j$.

$$\Delta\lambda_{1,-}^j = \frac{C_j^k + D_j^k}{-1 - \lambda_2 \sum_{l\in\mathcal{A}} \text{sgn}(\beta_l)L_{jl}\frac{\Delta\beta_l}{\Delta\lambda_1} - \sum_i (\sum_{l\in\mathcal{A}} x_{il}\frac{\Delta\beta_l}{\Delta\lambda_1})x_{ij}}$$

The step size $\Delta_{1,B}$ for the event B is:

$$\Delta\lambda_{1,B} = \max\{\Delta\lambda_{1,+}^j, \Delta\lambda_{1,-}^j : j \in \mathcal{A}^c; \Delta\lambda_{1,+}^j, \Delta\lambda_{1,-}^j \leq 0\}$$

After $\Delta\lambda_{1,A}$ and $\Delta\lambda_{1,B}$ are calculated, the final step size $\Delta\lambda_1$ can be obtained by:

$$\Delta\lambda_1 = \max(\Delta\lambda_{1,A}, \Delta\lambda_{1,B})$$

Now we can update the active set $\mathcal{A}$, λ_1,C_j and D_j. The next iteration $k+1$ consists of :solving the linear system (6) , calculating $\Delta\lambda_1$ and updating $\mathcal{A}$, λ_1,C_j and D_j . This entire process is repeated,until λ_1 reaches 0.

Between any two consecutive events, the solutions are linear in λ_1, and after an event occurs, the derivative of the solution with respect to λ_1 is changed. Therefore, the solution path is piecewise linear in λ_1, where each event corresponds to a kink on the path. The algorithm provides solutions at these kinks, and for any λ_1 between two consecutive kinks the solutions can be calculated precisely via linear interpolation. Following [2],to correct for potential bias due to double shrinkage,we adjust the estimate $\hat{\boldsymbol{\beta}}$ by a factor $1 + \lambda_2$.

Finally, if only training samples are available,10-fold cross-validation (CV) can be used for estimating the prediction error and for comparing models. For each fixed λ_2,we can use the number of steps for the lasso solution of the optimization problem as the second tuning parameter besides λ_2, which is selected by 10-fold CV.

IV. Properties of the proposed procedure

Given data set $(\boldsymbol{y}, \mathbf{X})$ and two fixed scalars (λ_1, λ_2),the response $\boldsymbol{y}$ is centered and predictors $\mathbf{X}$ are standardized. Let $\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)$ be the solution to equation (3). Suppose the two vertices i and j are only linked to each other on the network, $d_i = d_j = w(i, j)$. Then

$$||\hat{\boldsymbol{\beta}}_i(\lambda_1, \lambda_2)| - |\hat{\boldsymbol{\beta}}_j(\lambda_1, \lambda_2)|| \leq \frac{\|\boldsymbol{y}\|_1}{2\lambda_2}\sqrt{2(1 - \text{sgn}(\hat{\boldsymbol{\beta}}_i(\lambda_1, \lambda_2))\text{sgn}(\hat{\boldsymbol{\beta}}_j(\lambda_1, \lambda_2))\rho)}$$

where $||\boldsymbol{y}||_1 = \sum_{i=1}^n |y_i|$ and $\rho = \mathbf{x}_i^T\mathbf{x}_j$ is the sample correlation.

The proof of this theorem can be derived easily from [1] and [2]. The upper bound gives a quantitative description for the grouping effect of our proposed absolute value network-constrained regularization, which is half of the upper bound in the elastic net model. That means our regularization term has larger group effect than elastic net which is also indicated in Figure 1. Note that in [1] the group effect only exists in the first and third quadrants where $\hat{\boldsymbol{\beta}}_i(\lambda_1, \lambda_2)\hat{\boldsymbol{\beta}}_j(\lambda_1, \lambda_2) > 0$ which also coincides with Figure 1.

V. Simulation Results

Here we present the simulation results to demonstrate the performance of our proposed method which follows the simulation setup as [1] did. There are 200 transcription factors (TFs) in the simulated network and each TF regulates 10 genes. So the final network consists of 2200 genes and 2000 edges between the TFs and the corresponding regulated genes. We assume the first four TFs (active TFs) along with the genes they regulated contribute to the response $\boldsymbol{y}$ while the others are noise genes which are not related with response. The simulation data are generated by the following steps:

- The expression of the tth TF is generated according to $X_{TF_t} \sim N(0, 1)$ where $t = 1, \ldots, 200$
- The expression levels of the TF and the genes regulated by this TF are assumed to follow a bivariate normal distribution with correlation ρ. In [1] the correlation ρ is set as 0.7. To take account of the up-regulate or down-regulate function of each TF, we assign $\rho = 0.7$ or $\rho = -0.7$ with probability 0.5 for each gene. So conditioning on the expression level of the TF, the expression level of each gene it regulates follows a $N(\rho * X_{TF_t}, 0.51)$.
- The response $\boldsymbol{y}$ is generated by a linear regression model $y = \mathbf{X}\boldsymbol{\beta} + \varepsilon$ where $\varepsilon \sim N(0, \sum \beta_j^2/4)$.

Four models are generated according to different scenarios.

1)

$$\boldsymbol{\beta} = (5, \underbrace{\frac{5}{\sqrt{10}}, \ldots, \frac{5}{\sqrt{10}}}_{7}, \underbrace{\frac{-5}{\sqrt{10}}, \ldots, \frac{-5}{\sqrt{10}}}_{3},$$
$$-5, \underbrace{\frac{-5}{\sqrt{10}}, \ldots, \frac{-5}{\sqrt{10}}}_{7}, \underbrace{\frac{5}{\sqrt{10}}, \ldots, \frac{5}{\sqrt{10}}}_{3},$$
$$3, \underbrace{\frac{3}{\sqrt{10}}, \ldots, \frac{3}{\sqrt{10}}}_{7}, \underbrace{\frac{-3}{\sqrt{10}}, \ldots, \frac{-3}{\sqrt{10}}}_{3},$$
$$-3, \underbrace{\frac{-3}{\sqrt{10}}, \ldots, \frac{-3}{\sqrt{10}}}_{7}, \underbrace{\frac{3}{\sqrt{10}}, \ldots, \frac{3}{\sqrt{10}}}_{3}, 0, \ldots, 0)$$

This model reflects that among the 10 genes regulated by each active TF, the effects on the response of 7 genes are on the same direction (positive or negative) with the corresponding active TF while the effects of other 3 ones are not.

2)

$$\boldsymbol{\beta} = (5, \underbrace{\frac{5}{\sqrt{10}}, \ldots, \frac{5}{\sqrt{10}}}_{3}, \underbrace{\frac{-5}{\sqrt{10}}, \ldots, \frac{-5}{\sqrt{10}}}_{7},$$
$$-5, \underbrace{\frac{-5}{\sqrt{10}}, \ldots, \frac{-5}{\sqrt{10}}}_{3}, \underbrace{\frac{5}{\sqrt{10}}, \ldots, \frac{5}{\sqrt{10}}}_{7},$$
$$3, \underbrace{\frac{3}{\sqrt{10}}, \ldots, \frac{3}{\sqrt{10}}}_{3}, \underbrace{\frac{-3}{\sqrt{10}}, \ldots, \frac{-3}{\sqrt{10}}}_{7},$$
$$-3, \underbrace{\frac{-3}{\sqrt{10}}, \ldots, \frac{-3}{\sqrt{10}}}_{3}, \underbrace{\frac{3}{\sqrt{10}}, \ldots, \frac{3}{\sqrt{10}}}_{7}, 0, \ldots, 0)$$

This scenario assumes that only 3 genes that regulated by an active TF have same direction effect on the response while 7 genes impose differen direction effects from the TF.

3) The third and fourth models are adapted from model 1 and 2 by replacing the $\sqrt{10}$ in the denominators in $\boldsymbol{\beta}$ with 10, respectively.

A training set with 100 samples and an independent test set with 2000 samples are generated by simulation. The tuning parameters can be determined by 10-fold cross-validation. The simulation for each model is repeated 50 times. The prediction mean-squared errors (PMSE) on the test dataset are obtained. Sensitivity and specificity which indicate the ability of each method to select the relevant genes correctly are also calculated. Table I summarize the simulation results for these four different models. Our GEnet method gave smaller PMSEs compared with other methods. It is obvious that the PMSEs of network-constraint in model 2 and 4 are larger than that in model 1 and 3, respectively.

Table I
SIMULATION STUDY RESULTS BASED ON 50 SIMULATIONS AND STANDARD ERRORS ARE GIVEN IN PARENTHESES. HERE LASSO[4] ,ENET:ELASTIC NET [2]; NETWORK: NETWORK CONSTRAINT OF [1];GENET:THE PROPOSED METHOD.

	Model	1	2	3	4
Sensitivity	LASSO	0.21(0.12)	0.24(0.12)	0.22(0.07)	0.24(0.06)
	ENet	0.33(0.23)	0.38(0.21)	0.39(0.17)	0.37(0.16)
	Network	0.34(0.18)	0.35(0.19)	0.44(0.16)	0.34(0.13)
	GENet	0.37(0.21)	0.40(0.22)	0.54(0.19)	0.52(0.21)
Specificity	LASSO	0.31(0.18)	0.33(0.15)	0.34(0.13)	0.34(0.12)
	ENet	0.32(0.19)	0.34(0.16)	0.39(0.14)	0.37(0.16)
	Network	0.33(0.17)	0.32(0.18)	0.40(0.18)	0.35(0.14)
	GENet	0.34(0.17)	0.35(0.18)	0.46(0.18)	0.48(0.16)
PMSE	LASSO	87.9(13.6)	86.6(11.7)	33.4(4.6)	35.5(6.5)
	ENet	79.4(10.7)	78.1(9.4)	33.0(3.9)	35.2(6.6)
	Network	78.6(9.6)	79.9(9.6)	31.0(2.4)	33.7(4.5)
	GENet	75.9(9.6)	74.9(9.9)	30.4(2.4)	30.8(3.4)

This means the performance of network-constraint [1] will degenerate when there are more genes whose directions of effects on the respond are different from the TF which regulates them. The proposed GEnet is better in dealing with the fact that genes regulated by the same TF have both positive and negative effects on the response.

VI. REAL DATA ANALYSIS

The performance of the proposed method is evaluated in a microarray study of AD. The data set used here was generated from a microarray gene expression study of AD carried out by [10] which consists of hippocampal gene expression of 31 samples as well as MiniMental Status Examination (MMSE) scores of each sample. We then test the correlation of each gene's expression with MMSE scores across all 31 subjects in a linear regression model. These tests revealed several subnetworks that are related with the progression of AD. In our analysis, we select the regularization parameters λ_1 and λ_2 based on 10-fold cross-validation which are used on the 31 samples to identify the related subnetworks. The 31 samples are split randomly into training and testing sets for 100 times. In each split, 28 samples are used for training and the remaining 5 samples for testing. For the network term, we employ the interaction dataset which was obtained from EntrezGene and 33 human pathways in KEGG. We apply the network-constraint [1] and our proposed GEnet on the training data. Ten-folds cross-validation is used to select the tuning parameters λ_1 and λ_2.

Table II provides the results from the 50 experiments in term of prediction errors on the testing data, the number of genes selected based on the training data and the number of genes selected more than 24 times in the 50 experiments. We can see that the prediction performance and the number of selected genes of the two methods are similar. But our proposed GEnet method select more genes which are selected by $>$ 24 times in the 50 experiments than the

network constraint method which indicates more consistent genes are selected by our method. Further, our GEnet selects more genes among the genes selected by > 24 times than the network constraint method.

Fig. 2 depict the subnetworks with nodes larger than 3 identified by GEnet. In the top left subnetwork, HSPA2 and HSPA8 [11] can stabilize tau to support its binding to microtubules to avoid forming tangle. Uploading the genes in this subnetwork to functional annotation tools in DAVID Bioinformatics [12], we found the cell cycle pathway in KEGG is most enriched and recent work indicates Abeta oligomers induce neuronal cell cycle events in AD. The down left subnetwork in Fig. 2 involves amyloid precursor protein (APP), which gives rise to amyloid-β. It is well known that accumulation of amyloid-β peptides will form amyloid plaques ,which is the hallmark of AD. In the down right subnetwork, MDK interacts with STAT1 associating with JAK-STAT signaling pathway which affects the hyperphosphorylation of anomalous tau [13].

Table II
RESULTS ON 50 RANDOM SPLITS OF THE ORIGINAL DATA SETS.

Method	Network	GENet
Mean-squared error	66.9 ± 30.2	66.6 ± 29.5
number of selected genes	182.8 ± 82.3	186.4 ± 81.3
number of genes selected by $>$24 times	124	153

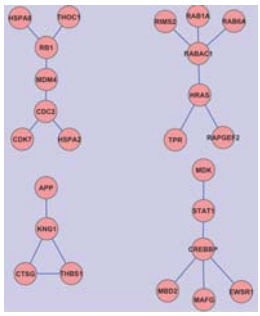

Figure 2. The identified sub-networks from the 153 most frequently selected genes by the proposed GEnet.

VII. CONCLUSION AND DISCUSSION

We have generalized the elastic net by incorporating prior network constraint on the absolute value of the coefficients. If we define $L(u, v) = 1$ when $u = v$ and $d_u = 0$ and all the nodes have no connections with one another, the Laplacian term $L = I$ and GENet will turn into the standard elastic net. So elastic net is the extreme instance of GENet with no nodes connected. The regularization term in [1] only impose the group effect in the first and third quadrants while our GENet expands the group effect to second and fourth quadrants.

An efficient optimization algorithm was developed for our GENet with the same spirit of LARS. Recently the coordinate descent attracts much attention. We will explore this new optimization procedure to solve our optimization problem in the future. And, we will use weighted network to replace the current binary network to address some potential errors in binary network. Finally, our optimization method can also be easily extended to classification.

REFERENCES

[1] C. Li and H. Li, "Network-constrained regularization and variable selection for analysis of genomic data," *Bioinformatics*, vol. 24, no. 9, pp. 1175–82, 2008.

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

[3] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, no. 2, pp. 407–451, 2004.

[4] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society Series B-Methodoloical*, vol. 58, no. 1, pp. 267–288, 1996.

[5] P. Wei and W. Pan, "Incorporating gene networks into statistical tests for genomic data via a spatially correlated mixture model," *Bioinformatics*, vol. 24, no. 3, pp. 404–11, 2008.

[6] W. Pan, B. Xie, and X. Shen, "Incorporating predictor network in penalized regression with application to microarray data," *Biometrics,To appear*, 2009.

[7] P. Zhao and B. Yu, "Boosted lasso," Journal of Machine Learning Research, Tech. Rep., 2004.

[8] F. R. K. Chung, *Spectral graph theory*. American Mathematical Society, 1997.

[9] S. Rosset and J. Zhu, "Piecewise linear regularized solution paths," *Annals of Statistics*, vol. 35, no. 3, pp. 1012–1030, 2007.

[10] E. M. Blalock, J. W. Geddes, K. C. Chen, N. M. Porter, W. R. Markesbery, and P. W. Landfield, "Incipient alzheimer's disease: microarray correlation analyses reveal major transcriptional and tumor suppressor responses," *Proc Natl Acad Sci USA*, vol. 101, no. 7, pp. 2173–8, 2004.

[11] F. Dou, W. J. Netzer, K. Tanemura, F. Li, F. U. Hartl, A. Takashima, G. K. Gouras, P. Greengard, and H. Xu, "Chaperones increase association of tau protein with microtubules," *Proc Natl Acad Sci U S A*, vol. 100, no. 2, pp. 721–6, 2003.

[12] W. Huang da, B. T. Sherman, and R. A. Lempicki, "Systematic and integrative analysis of large gene lists using david bioinformatics resources," *Nat Protoc*, vol. 4, no. 1, pp. 44–57, 2009.

[13] D. I. Orellana, R. A. Quintanilla, C. Gonzalez-Billault, and R. B. Maccioni, "Role of the jaks/stats pathway in the intracellular calcium changes induced by interleukin-6 in hippocampal neurons," *Neurotox Res*, vol. 8, no. 3-4, pp. 295–304, 2005.