Hindawi Publishing Corporation
Computational and Mathematical Methods in Medicine
Volume 2016, Article ID 3456153, 11 pages
http://dx.doi.org/10.1155/2016/3456153

Research Article

Efficient Regularized Regression with L_0 Penalty for Variable Selection and Network Construction

Zhenqiu Liu[1] and Gang Li[2]

[1]*Samuel Oschin Comprehensive Cancer Institute, Cedars-Sinai Medical Center, Los Angeles, CA 90048, USA*
[2]*Department of Biostatistics, School of Public Health, University of California at Los Angeles, Los Angeles, CA 90095-1772, USA*

Correspondence should be addressed to Zhenqiu Liu; liuzx@cshs.org

Received 6 April 2016; Revised 29 August 2016; Accepted 20 September 2016

Academic Editor: Xinyuan Song

Variable selections for regression with high-dimensional big data have found many applications in bioinformatics and computational biology. One appealing approach is the L_0 regularized regression which penalizes the number of nonzero features in the model directly. However, it is well known that L_0 optimization is NP-hard and computationally challenging. In this paper, we propose efficient EM (L_0EM) and dual L_0EM (DL_0EM) algorithms that directly approximate the L_0 optimization problem. While L_0EM is efficient with large sample size, DL_0EM is efficient with high-dimensional ($n \ll m$) data. They also provide a natural solution to all L_p $p \in [0, 2]$ problems, including lasso with $p = 1$ and elastic net with $p \in [1, 2]$. The regularized parameter λ can be determined through cross validation or AIC and BIC. We demonstrate our methods through simulation and high-dimensional genomic data. The results indicate that L_0 has better performance than lasso, SCAD, and MC+, and L_0 with AIC or BIC has similar performance as computationally intensive cross validation. The proposed algorithms are efficient in identifying the nonzero variables with less bias and constructing biologically important networks with high-dimensional big data.

1. Introduction

Variable selection with regularized regression has been one of the hot topics in machine learning and statistics. Regularized regressions identify outcome associated features and estimate nonzero parameters simultaneously and are particularly useful for high-dimensional big data with small sample sizes. Big data are datasets with either huge sample size or very high dimensions or both. In many real applications, such as bioinformatics, image and signal processing, and engineering, a large number of features are measured, but only a small number of features are associated with the dependent variables. Including irrelevant variables in the model will lead to overfitting and deteriorate the prediction performance. Therefore, different regularized regression methods have been proposed for variable selection and model construction. L_0 regularized regressions, which directly penalize the number of nonzero parameters, are the most essential sparsity measure. Several popular information criteria, including Akaike information criterion (AIC) [1], Bayesian information criterion (BIC) [2], and risk inflation criteria (RIC) [3], are based on L_0 penalty and have been used extensively for variable selections. However, solving a general L_0 regularized optimization is NP-hard and computationally challenging. Exhaustive search with AIC or BIC over all possible combinations of features is computationally infeasible with high-dimensional big data.

Different alternatives have been proposed for the regularized regression problem. One common approach is to replace L_0 by L_1. L_1 is known as the best convex relaxation of L_0. L_1 regularized regression [4] is convex and can be solved by an efficient gradient decent algorithm. It has found many applications in statistical genetics, bioinformatics, and medicine [5, 6]. Minimizing L_1 is equivalent to minimizing L_0 under certain conditions. However, the estimates of L_1 regularized regression are asymptotically biased, and lasso may not always choose the true model consistently [7]. Experimental results by Mancera and Portilla [8] also posed additional doubt about the equivalence of minimizing L_1 and L_0. Moreover, there were theoretical results [9] showing that while L_1 regularized regression never outperforms L_0 by a constant, in some cases L_1 regularized regression performs infinitely worse than L_0. Lin et al. [9] also showed that

the optimal L_1 solutions are often inferior to L_0 solutions found using greedy classic stepwise regression, although solutions with L_1 penalty can be found effectively. More recent approaches aimed to reduce bias and overcome discontinuity include SCAD [10], L_p $p \in (0,1]$ regularization [11, 12], and MC+ [13]. However, multiple free parameters (λ and p) must be tuned in those approaches, which is computationally intensive. They are not suitable for big data mining. Even though there are some effects for solving L_0 regularized optimization problems [14, 15], L_0 was either approximated by a different continuous smooth function or transformed into a much larger ranking optimization problem. To the best of our knowledge, currently, there is no efficient method directly approximating L_0 for big data problem.

In this paper, we propose efficient EM algorithms that directly approximate L_0 regularized regression problem. Our proposed approaches effectively deal with L_0 optimization by solving a sequence of convex L_2 optimizations and are efficient for high-dimensional data. They also provide a natural solution to all L_p $p \in [0,2]$ problems, including lasso with $p = 1$, elastic net with $p \in [1,2]$ [16], and the combination of L_1 and L_0 with $p \in (0,1]$ [17]. Similar to lasso, the regular parameter λ can be determined by the generalized information criterion [18]; optimal λ with L_0 regularized regression can also be predetermined with different model selection criteria such as AIC, BIC, and RIC. L_0 local graphical model with either AIC or BIC is faster than L_1 with cross validation. We demonstrate our methods through simulation and high-dimensional genomic data. The proposed methods identify the nonzero variables with less bias and outperform lasso, SCAD, and MC+ by a large margin. They can also choose the important genes and construct biological networks effectively.

2. Methods

Given an $n \times 1$ dependent variable $\mathbf{y}$ and an $n \times m$ feature matrix X, a linear model is defined as

$$\mathbf{y} = X\theta + \varepsilon, \tag{1}$$

where n is the number of samples and m is the number of variables and $n \ll m$, $\theta = [\theta_1, \ldots, \theta_m]^t$ are m parameters to be estimated, and $\varepsilon \sim N(0, \sigma^2 I_n)$ are the random errors with mean 0 and variance σ^2. Assume that only a small subset of $\{\mathbf{x}_j\}_{j=1}^m$ has nonzero θ_js. Let $R \subseteq \{1, \ldots, m\}$ be the subset index of relevant variables with $\theta_j \neq 0$, and let $O \subseteq \{1, \ldots, m\}$ be the index of irrelevant features with 0 coefficients; we have $R \cup O = \{1, 2, \ldots, m\}$, $X_R \cup X_O = X$, and $\theta_R \cup \theta_O = \theta$, where $\theta_O = 0$. The error function for L_0 regularized regression is

$$\begin{aligned} E &= \frac{1}{2}\left\|\mathbf{y} - X\theta\right\|^2 + \frac{\lambda}{2}\|\theta\|_0 \\ &= \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{m}\theta_j x_{ij}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{m} I\left(\theta_j \neq 0\right), \end{aligned} \tag{2}$$

where $\|\theta\|_0 = \sum_{j=1}^m I(\theta_j \neq 0) = |R|$ counts the number of nonzero parameters. One observation is that (2) is equivalent to (3) when reaching the optimal solution.

$$\begin{aligned} E &= \frac{1}{2}\left\|\mathbf{y} - X\theta\right\|^2 + \frac{\lambda}{2}\|\theta\|_0 = \frac{1}{2}\left\|\mathbf{y} - X\theta\right\|^2 + \frac{\lambda}{2}\sum_{j\in R} 1 \\ &= \frac{1}{2}\left\|\mathbf{y} - X\theta\right\|^2 + \frac{\lambda}{2}|R|. \end{aligned} \tag{3}$$

Our L_0EM methods will be derived from (3). We can rewrite (3) as the following two equations:

$$E = \frac{1}{2}\left\|\mathbf{y} - X\theta\right\|^2 + \frac{\lambda}{2}\sum_{j\in R}\frac{\theta_j^2}{\eta_j^2}, \tag{4}$$

$$\eta = \theta. \tag{5}$$

Given η_j, (4) is a convex quadratic function and can be optimized by taking the first-order derivative:

$$\nabla E = \lambda\theta_R \oslash \eta_R^2 - X_R^t\left(\mathbf{y} - X\theta\right) = 0, \tag{6}$$

where $\oslash$ indicates element-wise division. Rewriting (6), we have

$$\lambda\theta_R - \eta_R^2 \odot X_R^t\left(\mathbf{y} - X\theta\right) = 0. \tag{7}$$

In addition,

$$\lambda\theta_O - \eta_O^2 \odot X_O^t\left(\mathbf{y} - X\theta\right) = 0, \quad \forall\lambda > 0, \tag{8}$$

since $\theta_O = \eta_0 = 0$, where $\odot$ is element-wise multiplication, $\eta_R^2 \odot X_R^t = [\eta_R^2 \odot \mathbf{x}_{1R}^t, \ldots, \eta_R^2 \odot \mathbf{x}_{nR}^t]$, and $\eta_O^2 \odot X_O^t = [\eta_O^2 \odot \mathbf{x}_{1O}^t, \ldots, \eta_O^2 \odot \mathbf{x}_{nO}^t] = \mathbf{0}$. Let $D = \mathrm{diag}(\eta_1^2, \ldots, \eta_m^2)$ be a diagonal matrix with η_j^2s on the diagonal and combine (7) and (8) together; we have

$$\begin{aligned} \eta^2 \odot \nabla E &= \lambda\theta - DX^t\left(\mathbf{y} - X\theta\right) = \lambda\theta - DX^t\mathbf{y} + DX^tX\theta \\ &= 0. \end{aligned} \tag{9}$$

Solving (9) leads to following explicit solution:

$$\theta = \left(DX^tX + \lambda I\right)^{-1} DX^t\mathbf{y}, \tag{10}$$

$$\eta = \theta, \tag{11}$$

where (10) can be considered as the M-step of the EM algorithm maximizing negative cost function $-E$ and (11) can be regarded as the E-step with $E(\eta) = \theta$. Equations (10) and (11) together can also be treated as a fixed point iteration method in nonlinear optimization. It is guaranteed to have optimal solutions under certain conditions as shown in Theorem 1.

Theorem 1. *Given an input matrix X, output matrix $\mathbf{y}$, and initialized solution θ^0, the nonlinear system determined by (10) and (11) will converge to an optimal solution, when $\lambda\|(DX^tX + \lambda I)^{-2}\|_\infty\|\sqrt{D}X^t\mathbf{y}\|_\infty < 1/2$.*

Proof. Equations (10) and (11) are the same as

$$\begin{aligned}\theta &= \left(DX^tX + \lambda I\right)^{-1} DX^t\mathbf{y} \\ &= \left(\theta^2 \odot X^tX + \lambda I\right)^{-1}\left(\theta^2 \odot X^t\right)\mathbf{y}.\end{aligned} \tag{12}$$

First, $G(\theta) = (\theta^2 \odot X^tX + \lambda I)^{-1}(\theta^2 \odot X^t)\mathbf{y}$ is Lipschitz continuous for $\theta \in R^m$, and

$$\begin{aligned}\nabla G(\theta) &= \left(DX^tX + \lambda I\right)^{-2} \\ &\cdot \left[\left(\theta^2 \odot X^tX + \lambda I\right)\left(2\theta \odot X^t\right)\mathbf{y} - 2\theta\right. \\ &\left.\odot X^tX\left(\theta^2 \odot X^t\right)\mathbf{y}\right] = \left(\theta^2 \odot X^tX + \lambda I\right)^{-2}\left[2\lambda\theta\right. \\ &\left.\odot X^t\mathbf{y}\right] = 2\lambda\left(DX^tX + \lambda I\right)^{-2}\left(\sqrt{D}X^t\mathbf{y}\right).\end{aligned} \tag{13}$$

Because $\lambda\|(DX^tX + \lambda I)^{-2}\|_\infty\|\sqrt{D}X^t\mathbf{y}\|_\infty < 1/2$, it is clear from (13) that there is a Lipschitz constant

$$\begin{aligned}\gamma &= \|\nabla G(\theta)\|_\infty = 2\lambda 2\left\|\left(DX^tX + \lambda I\right)^{-2}\left(\sqrt{D}X^t\mathbf{y}\right)\right\|_\infty \\ &\le 2\lambda\left\|\left(DX^tX + \lambda I\right)^{-2}\right\|_\infty\left\|\sqrt{D}X^t\mathbf{y}\right\|_\infty < 2\cdot\frac{1}{2} = 1.\end{aligned} \tag{14}$$

So the derivative for every θ_j is less than 1. Now, given the initial value for (10) and (11) $\eta = \theta^0 \in R^m$, the sequence $\{\theta^r\}$ remains bounded because, $\forall i = 1, \ldots, r$,

$$\begin{aligned}\left\|\theta^{i+1} - \theta^i\right\|_\infty &= \left\|G\left(\theta^i\right) - G\left(\theta^{i-1}\right)\right\|_\infty \\ &= \left\|\nabla G(\xi)\left(\theta^i - \theta^{i-1}\right)\right\|_\infty \\ &\le \|\nabla G(\xi)\|_\infty\left\|\theta^i - \theta^{i-1}\right\|_\infty \\ &\qquad \left(\text{where } \xi \in \left(\theta^{i-1}, \theta^i\right)\right) \\ &= \gamma\left\|\theta^i - \theta^{i-1}\right\|_\infty \le \cdots \le \gamma^i\left\|\theta^1 - \theta^0\right\|_\infty.\end{aligned} \tag{15}$$

And therefore

$$\begin{aligned}\left\|\theta^r - \theta^0\right\|_\infty &= \left\|\sum_{i=0}^{r-1}\left(\theta^{i+1} - \theta^i\right)\right\|_\infty \le \left\|\theta^1 - \theta^0\right\|_\infty\sum_{i=0}^{r-1}\gamma^i \\ &\le \frac{\left\|\theta^1 - \theta^0\right\|_\infty}{(1-\gamma)}.\end{aligned} \tag{16}$$

Now, $\forall r, k \ge 0$,

$$\begin{aligned}\left\|\theta^{r+k} - \theta^r\right\|_\infty &= \left\|G\left(\theta^{r+k-1}\right) - G\left(\theta^{r-1}\right)\right\|_\infty \\ &\le \gamma\left\|\theta^{r+k-1} - \theta^{r-1}\right\|_\infty \\ &\le \gamma\left\|G\left(\theta^{r+k-2}\right) - G\left(\theta^{r-2}\right)\right\|_\infty \\ &\le \gamma^2\left\|\theta^{r+k-2} - \theta^{r-2}\right\|_\infty \le \cdots \\ &\le \gamma^r\left\|\theta^k - \theta^0\right\|_\infty \le \frac{\gamma^r\left\|\theta^1 - \theta^0\right\|_\infty}{1-\gamma}.\end{aligned} \tag{17}$$

Hence,

$$\lim_{r,k\to\infty}\left\|\theta^{r+k} - \theta^r\right\|_\infty = 0, \tag{18}$$

and therefore $\{\theta^r\}$ is a Cauchy sequence that has a limit solution θ^*.

Note that $G(\theta)$ is not a convex function; multiple local optimal solutions may exist. However, given initial θ^0, our algorithm always reaches the same optimal solution closest to θ^0. Assuming that there were two solutions θ^* and $\theta^\diamond$,

$$\left\|\theta^* - \theta^\diamond\right\|_\infty = \left\|G\left(\theta^*\right) - G\left(\theta^\diamond\right)\right\|_\infty \le \gamma\left\|\theta^* - \theta^\diamond\right\|_\infty. \tag{19}$$

Since $\gamma < 1$, (19) can only hold, if $\|\theta^* - \theta^\diamond\|_\infty = 0$. That is, $\theta^* = \theta^\diamond$, so the optimal solution of the EM algorithm is always the same.

Finally, the EM algorithm will be closer to the optimal solution at each step, because

$$\left\|\theta^{r+1} - \theta^*\right\|_\infty = \left\|G\left(\theta^r\right) - G\left(\theta^*\right)\right\|_\infty \le \gamma\left\|\theta^r - \theta^*\right\|_\infty. \tag{20}$$

□

Theorem 1 indicates that both the regularized parameter λ and initial values of the parameter θ are important. Given an initial value θ^0, the method converges to an optimal solution as long as $\lambda\|(DX^tX + \lambda I)^{-2}\|_\infty\|\sqrt{D}X^t\mathbf{y}\|_\infty < 1/2$.

Lemma 2. *When* $\lambda < \|DX^tX\|_\infty$ *and* $\|DX^tX\|_\infty > (1/2)\|\sqrt{D}X^t\mathbf{y}\|_\infty$, *the algorithm will find a nontrivial optimal solution for* θ. *More specifically, it will converge to an optimal solution, when* $\lambda < (1/4)\|(X^tX)^{-1}\mathrm{diag}^2(X^t\mathbf{y})\|_\infty$ *and* $\|\theta\|_\infty > (1/2)\|X^tX\|_\infty^{-1}\|X^t\mathbf{y}\|_\infty$ *for* λ *and* θ, *respectively, where* $\mathrm{diag}(\mathbf{x})$ *is a diagonal matrix with* x_i *on the diagonal.*

Proof. Since $\lambda\|(DX^tX + \lambda I)^{-2}\|_\infty\|\sqrt{D}X^t\mathbf{y}\|_\infty < 1/2$, we have

$$\begin{aligned}1 &> 2\lambda\left\|\left(DX^tX + \lambda I\right)^{-2}\right\|_\infty\left\|\sqrt{D}X^t\mathbf{y}\right\|_\infty \\ &\ge 2\lambda\left\|DX^tX + \lambda I\right\|_\infty^{-2}\left\|\sqrt{D}X^t\mathbf{y}\right\|_\infty \\ &\ge 2\lambda\left(\left\|DX^tX\right\|_\infty + \lambda\right)^{-2}\left\|\sqrt{D}X^t\mathbf{y}\right\|_\infty \\ &= \frac{2\lambda}{\left(\|DX^tX\|_\infty + \lambda\right)}\frac{\left\|\sqrt{D}X^t\mathbf{y}\right\|_\infty}{\left(\|DX^tX\|_\infty + \lambda\right)}.\end{aligned} \tag{21}$$

Inequality (21) is satisfied, if

$$\begin{aligned}&\frac{2\lambda}{\left(\|DX^tX\|_\infty + \lambda\right)} < 1, \\ &\frac{\left\|\sqrt{D}X^t\mathbf{y}\right\|_\infty}{\left(\|DX^tX\|_\infty + \lambda\right)} < 1 \\ &\qquad\Downarrow \\ &\lambda \le \left\|DX^tX\right\|_\infty, \\ &\left\|DX^tX\right\|_\infty > \frac{1}{2}\left\|\sqrt{D}X^t\mathbf{y}\right\|_\infty.\end{aligned} \tag{22}$$

In addition, we have

$$\begin{aligned}\left\|DX^tX\right\|_\infty &= \left\|\operatorname{diag}(\theta)\operatorname{diag}\left(\theta^t\right)X^tX\right\|_\infty \\ &= \left\|\operatorname{diag}\left(\left(DX^tX+\lambda I\right)^{-1}DX^t\mathbf{y}\right)\right. \\ &\quad\left.\cdot\operatorname{diag}\left(\mathbf{y}^tXD\left(DX^tX+\lambda I\right)^{-1}\right)X^tX\right\|_\infty \\ &\geq \left\|\left(2X^tX\right)^{-1}\operatorname{diag}^2\left(X^t\mathbf{y}\right)\left(2X^tX\right)^{-1}X^tX\right\|_\infty \\ &= \frac{1}{4}\left\|\left(X^tX\right)^{-1}\operatorname{diag}^2\left(X^t\mathbf{y}\right)\right\|_\infty,\end{aligned} \tag{23}$$

and let

$$\begin{aligned}\|D\|_\infty\left\|X^tX\right\|_\infty \geq \left\|DX^tX\right\|_\infty &> \frac{1}{2}\left\|\sqrt{D}\right\|_\infty\left\|X^t\mathbf{y}\right\|_\infty \\ &\geq \frac{1}{2}\left\|\sqrt{D}X^t\mathbf{y}\right\|_\infty.\end{aligned} \tag{24}$$

Therefore, if we take

$$\begin{aligned}\lambda &< \frac{1}{4}\left\|\left(X^tX\right)^{-1}\operatorname{diag}^2\left(X^t\mathbf{y}\right)\right\|_\infty \leq \left\|DX^tX\right\|_\infty, \\ \|\theta\|_\infty &= \left\|\sqrt{D}\right\|_\infty > \frac{1}{2}\left\|X^tX\right\|_\infty^{-1}\left\|X^t\mathbf{y}\right\|_\infty,\end{aligned} \tag{25}$$

the algorithm will find a nontrivial optimal solution. In particular, when $X^tX = I$, we have

$$\begin{aligned}\lambda &< \frac{1}{4}\left\|\operatorname{diag}^2\left(X^t\mathbf{y}\right)\right\|_\infty = \frac{1}{4}\max\left\{\left(\mathbf{x}_j^t\mathbf{y}\right)^2\right\}_{j=1}^m, \\ \|\theta\|_\infty &> \frac{1}{2}\left\|X^t\mathbf{y}\right\|_\infty.\end{aligned} \tag{26}$$

Both Theorem 1 and Lemma 2 provide some useful guidance for implementing the method and choosing appropriate λ and θ^0. They show that the EM algorithm always converges to an optimal solution, given certain λ and initial solution θ^0, and the estimated value is closer to the true solution after each EM iteration. Note that there is a trivial solution $\theta_j = 0, \forall j = 1,\ldots,m$, for (10) and (11); therefore, nonzero initial θ^0 is critical for finding a nontrivial solution. Our experiences with this method indicate that initializing θ with the estimates from L_2 penalized ridge regression will lead to quick convergence and super performance. The algorithm with such initialization usually converges under one hundred iterations and the performance is substantially better than lasso as shown in Section 3. The EM algorithm is as shown in Algorithm 1.

To deal with big data problem with $n \ll m$, we also propose an efficient algorithm by solving the much smaller $(n \times n)$ matrix inverse problem similar to [19]. The algorithm is based on the following fact:

$$\left(DX^tX+\lambda I_m\right)^{-1}DX^t = DX^t\left(XDX^t+\lambda I_n\right)^{-1}. \tag{27}$$

So θ has the analytical solution:

$$\theta = DX^t\left(XDX^t+\lambda I_n\right)^{-1}\mathbf{y}. \tag{28}$$

Given a $0 < \lambda \leq \lambda_{\max}$, a small number $\varepsilon = 1e-6$, and training data $\{X, \mathbf{y}\}$,
Initializing $\theta = (X^tX+\lambda I)^{-1}X^t\mathbf{y}$,
While 1,
  E-step: $\eta = \theta$, and $D = \operatorname{diag}(\eta_1^2,\ldots,\eta_m^2)$
  M-step: $\theta = (DX^tX+\lambda I)^{-1}DX^t\mathbf{y}$
  if $\|\theta-\eta\| < \varepsilon$, Break; End
End

Algorithm 1: L_0EM algorithm.

Given a $0 < \lambda \leq \lambda_{\max}$, a small number $\varepsilon = 1e-6$, and training data $\{X, \mathbf{y}\}$,
Initializing $\theta = X^t(XX^t+\lambda I_n)^{-1}\mathbf{y}$,
While 1,
  E-step: $\eta = \theta$, and $D = \operatorname{diag}(\eta_1^2,\ldots,\eta_m^2)$
  M-step: $\theta = DX^t(XDX^t+\lambda I_n)^{-1}\mathbf{y}$
  if $\|\theta-\eta\| < \varepsilon$, Break; End
End

Algorithm 2: DL_0EM algorithm.

The dual L_0EM (DL_0EM) algorithm dealing with $n \ll m$ problem with (28) is as shown in Algorithm 2.

Apparently, while L_0EM algorithm is efficient for solving the large n big data problem, DL_0EM can handle $n \ll m$ problem more efficiently. □

Lemma 3. *Given appropriate initial θ^0, the final solution of L_0EM and DL_0EM algorithm is an optimal solution for L_0 approximation problem that minimizes $E = (1/2)\|\mathbf{y}-X\theta\|^2 + (\lambda/2)|R|$ in (3).*

Proof. First, we show that the proposed algorithm is L_0 approximation. Given a high-dimensional matrix $X_{n\times m}(n \ll m)$ and a threshold γ for the coefficient estimates, L_0 rejects all the coefficient estimates below γ to 0 and keeps the large coefficients unchanged. This is the same as defining a binary vector $S = [0,\ldots,1,\ldots,1]^t$, with the value of 0 or 1 for each feature, where $S_j = 1$, if the coefficient estimate for that feature is above the threshold γ and 0 otherwise. Let $S = \operatorname{diag}(S)$ be a diagonal matrix with S on its diagonal; we have the selected feature matrix $X_S = XS$. We can build the standard models with the matrix X_S, if we know S in advance. For instance, we can estimate the coefficients of a ridge regression given X_S and y with

$$\begin{aligned}\theta &= \left(X_S^tX_S+\lambda I\right)^{-1}X_S^ty = \left(X_S^tX+\lambda I\right)^{-1}X^tSy \\ &= \left(SX^tX+\lambda I\right)^{-1}SX^ty,\end{aligned} \tag{29}$$

where $X_S^tX_S = SX^tXS = SX^tX$ because of the special structure of matrix S. It is guaranteed that the estimate for feature j is 0 with $S_j = 0$. However, in reality, we do not know S. Estimating both S and θ is NP-hard problem, since we need to solve a mixed-integer optimization problem.

Comparing (29) with the M-step of the primal algorithm, $\theta = (DX^tX + \lambda I_m)^{-1}DX^ty$, where $D = \text{diag}(\eta_1^2, \ldots, \eta_m^2)$; it is clear that S is replaced by D and binary S_j is approximated by continuous η_j^2 in the proposed algorithm. Therefore, The proposed algorithm is a direct L_0 approximation.

Next, we show that the proposed algorithm leads to a sparse solution. Note that the penalties for L_0 regularized regression in (4) are inversely proportional to the squared magnitude of the parameters. That is,

$$\lambda_j = \begin{cases} \dfrac{\lambda}{2\eta_j^2} & \text{if } \eta_j \neq 0 \\ \infty & \text{if } \eta_j = 0, \end{cases} \tag{30}$$

and $\eta = \theta$, when L_0EM or DL_0EM algorithm converges. Equation (30) shows that when the true parameter $\theta_j = 0$, the penalty λ_j goes to infinity, so $\widehat{\theta}_j$ must be 0 with the proposed algorithms. In addition, when the true parameters $\theta_j \neq 0$,

$$E = \frac{1}{2}\left\|\mathbf{y} - X\theta\right\|^2 + \frac{\lambda}{2}\sum_{j\in R}\frac{\theta_j^2}{\eta_j^2} = \frac{1}{2}\left\|\mathbf{y} - X\theta\right\|^2 + \frac{\lambda}{2}|R|, \tag{31}$$

because $\eta_j = \theta_j$, when the algorithm converges. Therefore, Lemma 3 holds. Note that our proposed methods will find a sparse solution with a large number of iterations and small ε, even though the solution of L_2 regularized regression is not sparse. Small parameters (θ_js) become smaller at each iteration and will eventually go to zero (below the machine epsilon). We can also set a parameter to 0 if it is below predefined $\varepsilon = 10e - 6$ to speed up the convergence of the algorithm.

The proposed algorithms are similar to the iteratively reweighted least square approach for L_p/L_q optimization in the literature [20, 21]. However, instead of solving L_p optimization problem directly, they added a small value ε in $\theta_j^2/(\eta_j^{2-p} + \varepsilon)$ to handle the undefined 0/0 problem when $\theta_j = 0$, leading to approximation and bias estimations. In our proposed algorithm, 0s are multiplied into the feature matrix X ($X_D = XD$). There is no undefined 0/0 problem in the proposed algorithm. Finally, similar procedures can be extended to general L_p; $p \in [0, 2]$ without much difficulty. L_p based EM algorithm L_pEM and the statistical properties of L_0 penalized regression are reported in the Appendix in Supplementary Material available online at http://dx.doi.org/10.1155/2016/3456153. The proposed L_pEM algorithm is similar to adaptive lasso [7] in that both use a weighted penalty. However, the weights in adaptive lasso are predetermined by ordinary least estimates when $n > m$ and univariate regression coefficients when $n < m$, which may lead to inaccurate estimate. In contrary, our proposed L_pEM updates the weights with an analytical solution at each iteration and automatically finds the optimal weights and estimates. □

L_0 Based Local Graphical Model. One important application of L_0 regularized regression is to detect high-order correlation structures, which have numerous real-world applications including gene network analysis. Given matrix X, let $\mathbf{x}_j$ be the jth variable, and let X_{-j} be the remaining variables; we have $P(\mathbf{x}_j \mid X_{-j}) \sim N(X_{-j}\theta, \sigma^2)$, where the coefficients θ measure the partial correlations between $\mathbf{x}_j$ and the rest of variables. Therefore, we will minimize

$$\arg\min_{\theta} E(\theta) = \arg\min_{\theta}\left\{\left\|\mathbf{x}_j - X_{-j}\theta\right\|^2 + \lambda\left\|\theta\right\|_0\right\}. \tag{32}$$

The high-order structure of X has been determined via a series of L_0 regularized regressions for each $\mathbf{x}_j$ with the remaining variables X_{-j}. The collected regression nonzero coefficients are the edges on the graph. L_0 local graphical model without cross validation is much efficient computationally than L_1 local graphical model. L_1 local graphical model is computationally intensive, because the regularized parameter λ for L_1 has to be determined through cross validation [22, 23]. For instance, given a matrix X with 100 variables, to find the optimal λ_{opt} from 100 candidate λ's with 5-fold cross validation, 500 models need to be evaluated for each variable $\mathbf{x}_j$. Therefore a total of 500×100 = 50000 models have to be estimated to detect the dependency among X with lasso. It usually takes hours to solve this problem. However, only 100 models are required to identify the same correlation structure with L_0 regularized regression and AIC or BIC, which only takes less than one minute to solve a similar problem. Notice that negative correlations between genes are difficult to confirm and seemingly less biologically relevant [24]. Most national databases are constructed with similarity (dependency) measures. It is straightforward to study only the positive dependency by simply setting θ ($\theta < 0$) = 0 in the EM algorithm.

Determination of λ. Regularized λ determines the sparsity of the model. The standard approach for choosing λ is cross validation and optimal λ is determined by the minimal mean squared error (MSE) of the test data (MSE = $\sum(y_i - \widehat{y}_i)^2/n$). One could also adapt the stability selection (SS) approach for λ determination [25, 26]. It chooses smallest λ that minimizes the inconsistencies in number of nonzero parameters with cross validation. We first calculate the mean and standard deviation (SD) of the number of nonzero parameters for each λ and then find smallest λ with SD = 0, where SD = 0 indicates that all models in k-fold cross validation have the same number of nonzero estimates. Our experiences indicate that larger λ chosen from both minimal MSE and stability selection ($\lambda = \max\{\lambda_{\text{MSE}}, \lambda_{\text{SS}}\}$) has the best performance. Choosing optimal λ from cross validation is computationally intensive and time-consuming. Fortunately, unlike lasso, identifying optimal λ for L_0 does not require using cross validation. Optimal λ_{opt} can be determined by variable selection criteria. Optimal λ_{opt} can be directly picked using AIC, BIC, or RIC criteria with $\lambda_{\text{opt}} = 2, \log n$, or $2\log m$, respectively. Each of these criteria is known to be optimal under certain conditions. This is a huge advantage of L_0, especially for big data problems. In general, we recommend to use BIC as information criteria for high-dimensional problem ($n \ll p$) and to use AIC when $n > p$.

Table 1: Performance measures for L_0 and L_1 regularized regression over 100 simulations, where values in the parentheses are the standard deviations. # SF: number of average selected features; MSE: average mean squared error; $\|\widehat{\theta} - \theta\|$: average absolute bias when comparing true and estimated parameters.

r	L_0			L_1		
	# SF	MSE	$\|\widehat{\theta} - \theta\|$	# SF	MSE	$\|\widehat{\theta} - \theta\|$
0	3.39 (±1.1)	1.01 (±0.14)	0.206 (±0.12)	14.5 (±3.45)	1.19 (±0.19)	0.38 (±0.1)
0.3	3.37 (±0.9)	1.02 (±0.16)	0.23 (±0.12)	14.5 (±2.91)	1.21 (±0.19)	0.41 (±0.19)
0.6	3.49 (±1.7)	1.02 (±0.23)	0.23 (±0.16)	13.5 (±3.0)	1.26 (±0.2)	0.54 (±0.15)
0.8	3.32 (±0.9)	1.06 (±0.15)	0.28 (±0.21)	11.7 (±2.69)	1.3 (±0.21)	0.89 (±0.25)

Table 2: Performance measures for L_0 and L_1 regularized regression with $\lambda = \max\{\lambda_{\text{MSE}}, \lambda_{\text{SS}}\}$ over 100 simulations, where values in the parenthesis are the standard deviations. # SF: number of average selected features; MSE: average mean squared error; $\|\widehat{\theta} - \theta\|$: average absolute bias when comparing true and estimated parameters.

r	L_0			L_1		
	# SF	MSE	$\|\widehat{\theta} - \theta\|$	#SF	MSE	$\|\widehat{\theta} - \theta\|$
0	3.09 (±0.53)	1.04 (±0.15)	0.18 (±0.11)	13.3 (±4.56)	1.21 (±0.17)	0.39 (±0.1)
0.3	3.08 (±0.54)	1.04 (±0.15)	0.17 (±0.07)	14.5 (±4.20)	1.22 (±0.17)	0.42 (±0.19)
0.6	3.10 (±0.46)	1.07 (±0.17)	0.21 (±0.10)	13.8 (±5.4)	1.27 (±0.47)	0.57 (±0.25)
0.8	3.02 (±0.14)	1.04 (±0.14)	0.26 (±0.13)	13.4 (±4.91)	1.25 (±0.21)	0.74 (±0.25)

3. Results

3.1. Simulation Study Application. To evaluate the performance of L_0 and L_1 regulation, we assume a linear model $\mathbf{y} = X\theta + \varepsilon$, where the input matrix X is from Gaussian distribution with mean $\mu = \mathbf{0}$ and different covariance structures Σ, where $\Sigma(i, j) = r^{|i-j|}$ with $r = 0, 0.3, 0.6, 0.8$, respectively. The true model is $\mathbf{y} = 2\mathbf{x}_1 - 3\mathbf{x}_2 + 4\mathbf{x}_5 + \varepsilon$ with $\varepsilon \sim N(0, 1)$. Therefore, only three features are associated with output $\mathbf{y}$, and the rest of θ_i's are zero. In our first simulation, we first compare L_0 and L_1 regularized regressions with a relatively small number of features $m = 50$ and a sample size of $n = 100$. Fivefold cross validation is used to determine optimal λ and compare the models performances. We seek to fit the regularized regression models over a range of regularization parameters λ. Each λ is chosen from $\lambda_{\min} = 1e - 4$ to $\lambda_{\max}$ with 100 equally log-spaced intervals, where $\lambda_{\max} = \max\{X^t\mathbf{y}\}$ for L_1 and $\max\{(\mathbf{x}_j^t\mathbf{y})^2/4\mathbf{x}_j^t\mathbf{x}_j\}$ for L_0. Lasso function in the statistics toolbox of MATLAB (http://www.mathworks.com/) is used for comparison. Cross validation with MSE is implemented nicely in the toolbox. The computational results are reported in Table 1. Table 1 shows that L_0 outperforms lasso in all categories by a substantial margin, when using the popular test MSE measure for model selection. In particular, the number of variables selected by L_0 are close to 3, the true number of nonzero variables, while lasso selected more than 11 features on average with different correlation structures ($r = 0, 0.3, 0.6, 0.8$). The test MSEs and bias both increase with the growth of correlation among features for both L_0 and lasso, but the test MSE and bias of L_0 are substantially lower than these of lasso. The maximal MSE of L_0 is 1.06, while the smallest MSE of L_1 is 1.19, and the largest bias of L_0 is 0.28, while the smallest bias of lasso is 0.38. In addition (results are not shown in Table 1), L_0 correctly identifies the true model 81, 74, 81, and 82 times for $r = 0, 0.3, 0.6$ and 0.8, respectively, over 100 simulations, while lasso never chooses the correct model. Therefore, compared to L_0 regularized regression, lasso selects more features than necessary and has larger bias in parameter estimation. Even though it is possible to get a correct model with lasso using larger λ, the estimated parameters will have a bigger bias and worse predicted MSE.

The same parameter setting is used for our second simulation, but the regularized parameter λ is determined by larger λ from both minimal MSE and stability selection ($\lambda = \max\{\lambda_{\text{MSE}}, \lambda_{\text{SS}}\}$). The computational results are reported in Table 2. Table 2 shows that the average number of associated features is much closer to 3 with slightly larger test MSEs. The maximal average number of features is 3.1 with $r = 0.6$, reduced from 3.49 with the test MSE only. In fact, with this combined model selection criteria and 100 simulations, L_0EM identified the true model with three nonzero parameters 95, 95, 95, and 97 times, respectively (not shown in the table), while lasso did not choose any correct models. The average bias of the estimates with L_0EM is also reduced. These indicate that the combination of test MSE and stability selection in cross validation leads to better model selection results than MSE alone with L_0EM. However, the computational results did not improve much with lasso. Over 13 features on average were selected under different correlation structures, suggesting that lasso inclines to select more spurious features than necessary. A much more conservative criterion with larger λ is required to select the right number of features, which will induce larger MSE and bias and deteriorate the prediction performance.

Simulation with High-Dimensional Data. Our third simulation deals with high-dimensional data with the number of samples $n = 100$ and the number of features $m = 1000$. The correlation structure is set to $r = 0, 0.3, 0.6$, and the same model $\mathbf{y} = 2\mathbf{x}_1 - 3\mathbf{x}_2 + 4\mathbf{x}_5 + \varepsilon$ was used for evaluating model

Table 3: Performance measures for L_0, L_1, SCAD, and MC+ regularized regressions with cross validation and $\lambda = \text{Max}\{\lambda_{\text{MSE}}, \lambda_{\text{SS}}\}$ over 100 simulations and the sample size of $n = 100$, and $m = 1000$, where values in the parenthesis are the standard deviations. # SF: number of average selected features; MSE: average mean squared error; $\|\widehat{\theta} - \theta\|$: average absolute bias when comparing true and estimated parameters.

	Measures	$r = 0$	$r = 0.3$	$r = 0.6$
L_0	# SF	3 (±0)	2.9 (±0.47)	2 (±0.73)
	$\|\widehat{\theta} - \theta\|$	0.14 (±0.09)	0.39 (±0.63)	1.69 (±1.25)
	Test MSE	1.14 (±0.34)	1.59 (±1.3)	2.8 (±1.72)
	# true model	100/100	78/100	23/100
L_1	# SF	24 (±18.4)	31.3 (±20.7)	36.7 (±16.5)
	$\|\widehat{\theta} - \theta\|$	0.57 (±0.11)	0.73 (±0.13)	1.14 (±0.25)
	Test MSE	1.50 (±0.25)	1.63 (±0.29)	1.92 (±0.41)
	# true model	0/100	0/100	0/100
SCAD	# SF	106.8 (±110.6)	73 (±111)	56.2 (±62.4)
	$\|\widehat{\theta} - \theta\|$	0.62 (±0.13)	0.72 (±0.14)	1.13 (±0.26)
	Test MSE	1.32 (±0.27)	1.54 (±0.27)	2.04 (±0.51)
	# true model	0/100	0/100	0/100
MC+	# SF	60.3 (±38.6)	70.5 (±26.0)	78.73 (±16.5)
	$\|\widehat{\theta} - \theta\|$	0.56 (±0.14)	0.66 (±0.12)	0.78 (±0.17)
	Test MSE	1.25 (±0.21)	1.31 (±0.27)	1.46 (±0.27)
	# true model	0/100	0/100	0/100

performance. Besides L_1, L_0 was also compared with two other cutting-edge regulations including SCAD and MC+, implemented in SparseReg package [27]. The simulation was repeated 100 times. The computational results are reported in Table 3. Table 3 shows that L_0 outperforms lasso by a large margin when correlations among features are low. When there is no correlation among features, 100 out of 100 simulations identify the true model with L_0, and 78 out of 100 simulations choose the correct model when $r = 0.3$, while lasso, SCAD, and MC+ choose more features than necessary and no true model was found under any correlation setting. However, when correlations among features are large with $r = 0.6$, the results are mixed. L_0 can still identify 23 out of 100 correct models, but the test MSE and bias of the parameter estimate of L_0 are slightly large than those of lasso, MC+, and SCAD. MC+ has the second best performance with less bias and smaller test MSE but chooses more features than necessary. In addition, we notice that L_0 is a more sparse model when correlation increases, indicating that L_0 tends to choose independent features. One reason for selecting more features with SCAD and MC+ may be that we only tuned the parameter λ and fixed $\gamma = 3.7$ and $\gamma = 1$ for SCAD and MC+, respectively. A regularization path of L_0 regression is shown in Figure 1. As shown in Figure 1(a), the three associated features first increase their values when λ goes larger and then go to zero when λ becomes extremely big, while the rest of the irrelevant features all go to zero when λ increases. Unlike lasso, which shrinks all parameters uniformly, L_0 will only force the estimates of irrelevant features to go to zero, while keeping the estimates of relevant features to their true value. This is the well-known oracle property of L_0. The oracle property means that the penalized estimator is asymptotically equivalent to the oracle estimator obtained only with signal variables without penalization. For this specific simulation, the three parameters $[\widehat{\theta_1}, \widehat{\theta_2}, \widehat{\theta_5}] = [1.85, -2.94, 4.0]$, very close to their true values $[2, -3, 4]$. Figures 1(b) and 1(c) are the test MSE and the standard deviation of the number of nonzero variables. Optimal λ is chosen from larger λ with minimal test MSE and stability selection as shown in the vertical lines of Figure 1.

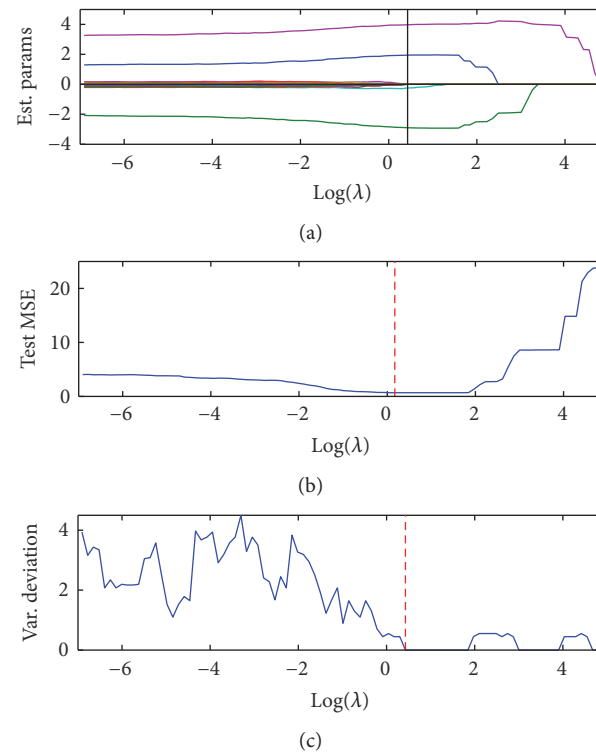

Figure 1: Regularized path for L_0 penalized regression with $n = 100$, $m = 1000$, and $r = 0.3$.

Table 4: Performance measures for L_0 regularized regression with AIC and BIC over 100 simulations with $n = 100$, and $m = 1000$, where values in the parenthesis are the standard deviations. # SF: number of average selected features; MSE*: in-sample average mean squared error; $\|\widehat{\theta} - \theta\|$: average absolute bias when comparing true and estimated parameters.

	Measures	$r = 0$	$r = 0.3$	$r = 0.6$
AIC	# SF	3.26 (±0.54)	3.72 (±1.94)	4.8 (±2.77)
	$\|\widehat{\theta} - \theta\|$	0.19 (±0.09)	0.36 (±0.58)	1.02 (±1.2)
	MSE*	0.96 (±0.14)	1.02 (±0.31)	1.27 (±0.51)
	# true model	78/100	73/100	59/100
BIC	# SF	3.0 (±0.0)	3.0 (±0.38)	2.89 (±0.80)
	$\|\widehat{\theta} - \theta\|$	0.16 (±0.08)	0.45 (±0.69)	1.80 (±1.20)
	MSE*	0.97 (±0.15)	1.29 (±0.81)	2.48 (±1.17)
	# true model	100/100	94/100	53/100

L_0 Regularized Regression without Cross Validation. Choosing the optimal parameter λ_{opt} with cross validation is time-consuming, especially with big data. As we mentioned previously, optimal λ can be picked from theory instead of cross validation. Since we are dealing with $n \ll m$ big data problem, RIC with $\lambda_{\text{opt}} = 2 \log m$ tends to penalize the parameters too much. So computational results with AIC and BIC without cross validation are reported in Table 4. Table 4 shows that L_0 regularized regression with AIC and BIC performs very well when compared with the results from computationally intensive cross validation in Table 3. Without correlation, BIC identifies the true model (100%), which is the same as cross validation in Table 3 and better than AIC's 78%. The bias of BIC (0.16) is only slightly higher than that of cross validation (0.14) but lower than that of AIC (0.19). Even though MSE*'s with AIC and BIC are in-sample mean squared errors, which are not comparable to the test MSE with cross validation, larger MSE* with BIC indicates that BIC is a more stringent criterion than AIC and selects less variables. With mild correlation ($r = 0.3$) and some sacrifices in bias and MSE*, BIC performs better than AIC in variable selection, since the average number of features selected is exactly 3 and 94% of the simulations recognize the true model, while AIC chooses more features (3.72) than necessary and only 73% of the simulations are right on targets. Cross validation is the most tight measure with 2.9 features on average and 75% of the simulations finding the correct model. When the correlations among the variables are high ($r = 0.6$), the results are mixed. Both BIC and AIC correctly identify more than half of the true models, while cross validation only recognizes 25% (5/20) of the model correctly. Therefore, compared with the computationally intensive cross validation, both BIC and AIC perform reasonably well. The performance of BIC is similar to cross validation with less computational time. In addition, we have suggested to use the result of ridge regression as the initial value for the proposed algorithms. However, the proposed algorithm is quite stable with different initializations. With $n = 100$, $p = 200$, $r = 0.3$, and 100 times of randomized initializations, the estimates of three nonzero parameters are $[\beta_1, \beta_2, \beta_5] = [2.05 \pm 0.08, -2.89 \pm 0.08, 4.01 \pm 0.09]$ with BIC criteria.

Simulations for Graphical Models. We simulate two network structures similar to those in Zhang and Mallick [28]: (i) band 1 network, where Σ is a covariance matrix with $\sigma_{ij} = 0.6^{|i-j|}$, so $A = \Sigma^{-1}$ has a band 1 network structure, and (ii) a more difficult problem for a band 2 network with weaker correlations, where $A = -\Sigma^{-1}$ with

$$a_{ij} = \begin{cases} 0.25 & \text{if } |i-j| = 1 \\ 0.4 & \text{if } |i-j| = 2 \\ 0 & \text{otherwise.} \end{cases} \quad (33)$$

The sample sizes are $n = 50$, 100, and 200, respectively, and the number of variables is $m = 100$. L_0 regularized regression with AIC and BIC is used to detect the network (correlation) structure. The consistency between the true and predicted structures is measured by the area under the ROC curve (AUC), false discovery (positive) rate (FDR/FPR), and false negative rate (FNR) of edges. The computational results are shown in Table 5. Table 5 shows that both AIC and BIC performed well. Both achieved at least 0.90 AUC for band 1 network and 0.8 AUC for band 2 network with different sample sizes. AIC performed slightly better than BIC, especially for band 2 network with weak correlations and small sample sizes. This is reasonable because BIC is a heavier penalty and forces most of the weaker correlations with $a_{ij} = 0.25$ to 0. In addition, BIC has slightly larger AUCs for band 1 network with strong correlation $r = 0.6$ and larger sample size ($n = 100, 200$). One interesting observation is that FDRs of both AIC and BIC are well controlled. Maximal FDRs of AIC for bands 1 and 2 networks are 0.29% and 0.2%, while maximal FDRs of BIC are only 0.1%, and 0.03%, respectively. Controlling false discovery rates is crucial for identifying true associations with high-dimensional data in bioinformatics. In general, AUC increases and both FDR and FNR decrease, as the sample sizes become larger, except for band 2 network with BIC. The performance of BIC is not necessarily better with a larger sample size, since the penalty λ increases with the sample size. L_1 graphical model was also used for comparison purpose [29, 30]. L_1 graphical model performed equally well as AIC and BIC with band 1 network but was the worst with the more difficult band 2 network. More interestingly, L_1 had the largest FDR, indicating that it selects more features than necessary.

3.2. Application to Real Ovarian Cancer Data. The purpose of this application is to identify subnetworks and study the

Table 5: Performance measures for L_0 regularized regression for graphical structure detection over 100 simulations, where values in the parenthesis are the standard deviations.

		Band 1			Band 2	
AIC	AUC	FDR (%)	FNR (%)	AUC	FDR (%)	FNR (%)
$n = 50$	.95 (±.01)	.29 (±.08)	9.4 (±2.6)	.82 (±.01)	.10 (±.05)	36.7 (±1.5)
100	.99 (±.005)	.20 (±.06)	1.2 (±1.1)	.84 (±.01)	.11 (±.04)	32.7 (±1.9)
200	.999 (±.0003)	.20 (±.05)	0 (±0)	.93 (±.01)	.11 (±.04)	14.2 (±2.4)
BIC	AUC	FPR (%)	FNR (%)	AUC	FPR (%)	FNR (%)
$n = 50$	.90 (±.02)	.10 (±.05)	20 (±3.6)	.803 (±.008)	.02 (±.02)	39.3 (±1.5)
100	.991 (±.007)	.03 (±.03)	1.8 (±1.3)	.83 (±.01)	.03 (±.02)	34.9 (±1.6)
200	.9999 (±.0005)	.01 (±.01)	.01 (±.10)	.82 (±.01)	.03 (±.02)	36.7 (±1.8)
L_1	AUC	FPR (%)	FNR (%)	AUC	FPR (%)	FNR (%)
$n = 50$	.91 (±.03)	3.5 (±.05)	11 (±3.6)	0.77 (±.01)	5.3 (±.07)	40.9 (±.62)
100	.99 (±.003)	1.52 (±.22)	.33 (±.67)	0.78 (±.007)	7.1 (±1.4)	36.3 (±1.1)
200	.99 (±.003)	1.21 (±.07)	.45 (±.53)	0.79 (±.01)	8.1 (±.57)	34.0 (±1.4)

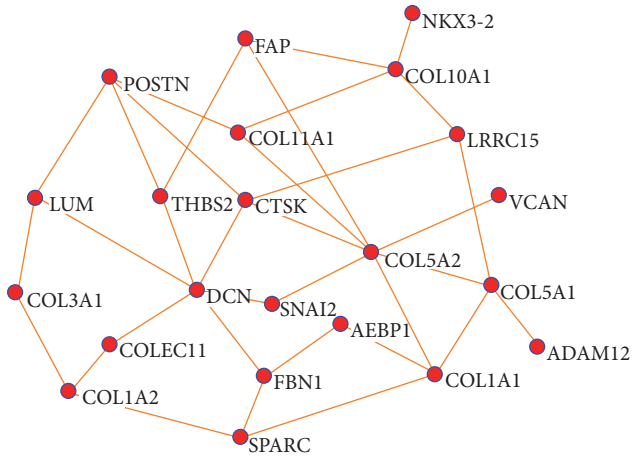

Figure 2: Subnetwork constructed with L_0 penalized regression, multisource gene expression profiling, and BIC.

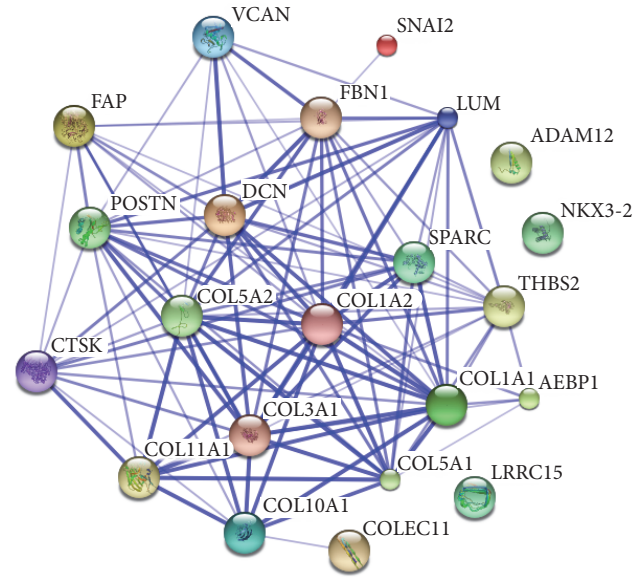

Figure 3: Known and predicted protein-protein interactions with the 22 genes on the subnetwork of Figure 2, where nodes represent proteins (genes) and edges indicate the direct (physical) and indirect (functional) associations. Stronger associations are represented by thicker lines.

biological mechanisms of potential prognostic biomarkers for ovarian cancer with multisource gene expression data. The ovarian cancer data was downloaded from the KMplot website (http://www.kmplot.com/ovar/) [31]. They originally got the data from searching Gene Expression Omnibus (GEO; http://www.ncbi.nlm.nih.gov/geo/) and The Cancer Genome Atlas (TCGA; http://cancergenome.nih.gov/) with multiple platforms. All collected datasets have raw gene expression data, survival information, and at least 20 patients available. They merged the datasets across different platforms carefully. The final data has 1287 patients samples and 22277 probe sets representing 13435 common genes. We identified 112 top genes that are associated with patient survival times using univariate Cox regression. We constructed a coexpression network from the 112 genes with L_0 regularized regression and identified biologically meaningful subnetworks (modules) associated with patient survival. Network is constructed with positive correlation only and BIC. The computational time for constructing such network is less than 2 seconds. One survival associated subnetwork we identified is given in Figure 2. The 22 genes on the subnetwork were then uploaded onto STRING (http://string-db.org/). STRING is an online database for exploring known and predicted protein-protein interactions (PPI). The interactions include direct (physical) and indirect (functional) associations. The predicted methods for PPI implemented in STRING include text mining, national databases, experiments, coexpression, cooccurrence, gene fusion, and neighborhood on the chromosome. The PPI networks for the 22 genes are presented in Figure 3. Comparing Figures 3 and 2, we conclude that the 22 identified genes on the subnetwork of Figure 2 are functioning together and have enriched important biological interactions and associations. Nineteen out of 22 genes on the survival associated subnetwork also have interactions on the known and predicted PPI network, except for genes LRRC15,

ADAM12, and NKX3-2. Even though they are not completely identical, many interactions on our subnetwork can also be verified on the PPI interaction network of Figure 3. For instance, collagen COL5A2 is the most important gene with the largest number of degrees (7) on our subnetwork. Six out of 7 genes that link to COL5A2 also have direct edges on the PPI network. Those direct connected genes (proteins) include FAP, CTSK, VCAN, COL1A1, COL5A1, and COL11A1. The remaining gene SNAI2 was indirectly linked to COL5A2 through FBN1 on the PPI network. In addition, one of the other important genes with the degree of the node (6) is Decorin (DCN). Four out of 6 genes directly connected to DCN on our subnetwork were confirmed on the PPI network, including FBN1, CTSK, LUM, and THBS2. The remaining two genes (SNAI2 and COLEC11) are indirectly connected to DCN on the PPI network. As indicated on Figure 2, the remaining 5 important genes with degree of node 4 are POSTN, CTSK, COL1A1, COL5A1, and COL10A1, and 8 genes with degree of node 3 are FBN1, LUM, LRRC15, COL11A1, THBS2, SPARC, COL1A2, and FAP, respectively. Furthermore, those 22 genes are involved in the biological process of GO terms, including extracellular matrix organization and disassembly and collagen catabolic, fibril, and metabolic processes. They are also involved in several important KEGG pathways including ECM-receptor interaction, protein digestion and absorption, amoebiasis, focal adhesion, and TGF-β signaling pathways. Finally, a large proportion of the 22 genes are known to be associated with poor overall survival (OS) in ovarian cancer. For instance, VCAN and POSTN were demonstrated *in vitro* to be involved in ovarian cancer invasion induced by TGF-β signaling [32], and COL11A1 was shown to increase continuously during ovarian cancer progression and to be highly overexpressed in recurrent metastases. Knockdown of COL11A1 reduces migration, invasion, and tumor progression in mice [33]. Other genes such as FAP, CTSK, FBN1, THBS2, SPARC, and COL1A1 are also known to be ovarian cancer associated [34–39]. Those genes contribute to cell migration and the progression of tumors and may be potential therapeutic targets for ovarian cancer, indicating that the proposed method can be used to construct biologically important networks efficiently.

4. Discussion

We proposed efficient EM algorithms for variable selection with L_0 regularized regression. The proposed algorithms find the optimal solutions of L_0 through solving a sequence of L_2 based ridge regressions. Given an initial solution, the algorithm will be guaranteed to converge to a unique solution under mild conditions, and the EM algorithm will be closer to the optimal solution after each iteration. Asymptotic properties, namely, consistency and oracle properties for exact L_0, are established under mild conditions. Our method applies to fixed, diverging, and ultra-high-dimensional problems with ten or hundred thousands of features. We compare the performance of L_0 regularized regression and lasso with simulated low- and high-dimensional data. L_0 regularized regression outperforms lasso, SCAD, and MC+ by a substantial margin under different correlation structures. Unlike lasso, which selects more features than necessary, L_0 regularized regression chooses the true model with high accuracy, less bias, and smaller test MSE, especially when the correlation is weak. Cross validation with the computation of the entire regularization path is computationally intensive and time-consuming. Fortunately L_0 regularized regression does not require it. Optimal λ_{opt} can be directly determined from AIC, BIC, and RIC. Those criteria are optimal under appropriate conditions. We demonstrate that both AIC and BIC performed well when compared to cross validation. Therefore, there is a big computational advantage of L_0, especially with big data. In addition, we demonstrate that L_0 regularized regression controls the false discovery (positive) rate (FDR) well with both AIC and BIC with the simulation of graphical models. The FDR is very low under different sample sizes with both AIC and BIC. Controlling FDR is crucial for biomarker discovery and computational biology, because further verifying the candidate biomarkers is time-consuming and costly. We applied our proposed method to construct a network for ovarian cancer from multisource gene expression data and identified a subnetwork that is important both biologically and clinically. We demonstrated that we can identify biologically important genes and pathways efficiently. Even though we demonstrated our method with gene expression data, the proposed method can be used for RNA-seq and metagenomic data, given that the data are appropriately normalized. Finally, because of the nonconvexity of L_0 regularized regression, there are multiple local optimal solutions for θ_j including a trivial solution $\theta_j = 0, \forall j = 1, \ldots, m$, as shown in (28). However, the nontrivial solution can be found efficiently as long as all parameters were initialized with nonzero values. We recommend the solution of ridge regression as an initial solution for the proposed algorithms.

Competing Interests

The authors declare that they have no competing interests.

References

[1] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, pp. 716–723, 1974.

[2] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[3] D. P. Foster and E. I. George, "The risk inflation criterion for multiple regression," *The Annals of Statistics*, vol. 22, no. 4, pp. 1947–1975, 1994.

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

[5] J. J. Li, C.-R. Jiang, J. B. Brown, H. Huang, and P. J. Bickel, "Sparse linear modeling of next-generation mRNA sequencing (RNA-Seq) data for isoform discovery and abundance estimation," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 50, pp. 19867–19872, 2011.

[6] Y. Li, M. Liang, and Z. Zhang, "Regression analysis of combined gene expression regulation in acute myeloid leukemia," *PLoS Computational Biology*, vol. 10, no. 10, Article ID e1003908, 2014.

[7] H. Zou, "The adaptive lasso and its oracle properties," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1418–1429, 2006.

[8] L. Mancera and J. Portilla, "L_0 norm based sparse representation through alternative projections," in *Proceedings of the International Conference on Image Processing (ICIP '06)*, 2006.

[9] D. Lin, D. P. Foster, and L. H. Ungar, "A risk ratio comparison of L_0 and L_1 penalized regressions," Tech. Rep., University of Pennsylvania, Philadelphia, Pa, USA, 2010.

[10] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[11] Z. Liu, S. Lin, and M. Tan, "Sparse support vector machines with L_p penalty for biomarker identification," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 7, no. 1, pp. 100–107, 2010.

[12] R. Mazumder, J. H. Friedman, and T. Hastie, "SparseNet: coordinate descent with nonconvex penalties," *Journal of the American Statistical Association*, vol. 106, no. 495, pp. 1125–1138, 2011.

[13] C.-H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *The Annals of Statistics*, vol. 38, no. 2, pp. 894–942, 2010.

[14] L. Dicker, B. Huang, and X. Lin, "Variable selection and estimation with the seamless-L_0 penalty," *Statistica Sinica*, vol. 23, pp. 929–962, 2013.

[15] Z. Lu and Y. Zhang, "Sparse approximation via penalty decomposition methods," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2448–2478, 2013.

[16] H. Zou and H. H. Zhang, "On the adaptive elastic-net with a diverging number of parameters," *The Annals of Statistics*, vol. 37, no. 4, pp. 1733–1751, 2009.

[17] Y. Liu and Y. Wu, "Variable selection via a combination of the L_0 and L_1 penalties," *Journal of Computational and Graphical Statistics*, vol. 16, no. 4, pp. 782–798, 2007.

[18] Y. Fan and C. Y. Tang, "Tuning parameter selection in high dimensional penalized likelihood," *Journal of the Royal Statistical Society—Series B: Statistical Methodology*, vol. 75, no. 3, pp. 531–552, 2013.

[19] Z. Liu, F. Sun, J. Braun, D. P. B. McGovern, and S. Piantadosi, "Multilevel regularized regression for simultaneous taxa selection and network construction with metagenomic count data," *Bioinformatics*, vol. 31, no. 7, pp. 1067–1074, 2015.

[20] M. J. Lai, Y. Y. Xu, and W. T. Yin, "Improved iteratively reweighted least squares for unconstrained smoothed ℓ_q minimization," *SIAM Journal on Numerical Analysis*, vol. 51, no. 2, pp. 927–957, 2013.

[21] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively re-weighted least squares minimization for sparse recovery," *Communications on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1–38, 2010.

[22] J. Peng, P. Wang, N. Zhou, and J. Zhu, "Partial correlation estimation by joint sparse regression models," *Journal of the American Statistical Association*, vol. 104, no. 486, pp. 735–746, 2009.

[23] Q. Liu and A. Ihler, "Learning scale free networks by reweighted L1 regularization," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS '11)*, Fort Lauderdale, Fla, USA, 2011.

[24] H. K. Lee, A. K. Hsu, J. Sajdak, J. Qin, and P. Pavlidis, "Coexpression analysis of human genes across many microarray data sets," *Genome Research*, vol. 14, no. 6, pp. 1085–1094, 2004.

[25] H. Liu, K. Roeder, and L. Wasserman, "Stability approach to regularization selection for high dimensional graphical models," in *Advances in Neural Information Processing Systems*, MIT Press, 2010.

[26] N. Meinshausen and P. Bhlmann, "Stability selection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 4, pp. 417–473, 2010.

[27] H. Zhou and Y. Wu, "A generic path algorithm for regularized statistical estimation," *Journal of the American Statistical Association*, vol. 109, no. 506, pp. 686–699, 2014.

[28] L. Zhang and B. K. Mallick, "Inferring gene networks from discrete expression data," *Biostatistics*, vol. 14, no. 4, pp. 708–722, 2013.

[29] Z. Liu, S. Lin, and S. Piantadosi, "Network construction and structure detection with metagenomic count data," *BioData Mining*, vol. 8, article 40, 2015.

[30] Z. Liu, S. Lin, N. Deng, D. McGovern, and S. Piantadosi, "Sparse inverse covariance estimation with L0 penalty for network construction with omics data," *Journal of Computational Biology*, vol. 23, no. 3, pp. 192–202, 2016.

[31] B. Gyorffy, A. Lánczky, and Z. Szállási, "Implementing an online tool for genomewide validation of survival-associated biomarkers in ovarian-cancer using microarray data from 1287 patients," *Endocrine-Related Cancer*, vol. 19, no. 2, pp. 197–208, 2012.

[32] T.-L. Yeung, C. S. Leung, K.-K. Wong et al., "TGF-β Modulates ovarian cancer invasion by upregulating CAF-Derived versican in the tumor microenvironment," *Cancer Research*, vol. 73, no. 16, pp. 5016–5028, 2013.

[33] D.-J. Cheon, Y. Tong, M.-S. Sim et al., "A collagen-remodeling gene signature regulated by TGF-β signaling is associated with metastasis and poor survival in serous ovarian cancer," *Clinical Cancer Research*, vol. 20, no. 3, pp. 711–723, 2014.

[34] M. Riester, W. Wei, L. Waldron et al., "Risk prediction for late-stage ovarian cancer by meta-analysis of 1525 patient samples," *Journal of the National Cancer Institute*, vol. 106, no. 5, Article ID dju048, 2014.

[35] G. Zhao, J. Chen, Y. Deng et al., "Identification of NDRG1-regulated genes associated with invasive potential in cervical and ovarian cancer cells," *Biochemical and Biophysical Research Communications*, vol. 408, no. 1, pp. 154–159, 2011.

[36] W. Zhang, T. Ota, V. Shridhar, J. Chien, B. Wu, and R. Kuang, "Network-based survival analysis reveals subnetwork signatures for predicting outcomes of ovarian cancer treatment," *PLoS Computational Biology*, vol. 9, no. 3, Article ID e1002975, 2013.

[37] N. L. Gardi, T. U. Deshpande, S. C. Kamble, S. R. Budhe, and S. A. Bapat, "Discrete molecular classes of ovarian cancer suggestive of unique mechanisms of transformation and metastases," *Clinical Cancer Research*, vol. 20, no. 1, pp. 87–99, 2014.

[38] L. Tang and J. Feng, "SPARC in tumor pathophysiology and as a potential therapeutic target," *Current Pharmaceutical Design*, vol. 20, no. 39, pp. 6182–6190, 2014.

[39] P.-N. Yu, M.-D. Yan, H.-C. Lai et al., "Downregulation of miR-29 contributes to cisplatin resistance of ovarian cancer cells," *International Journal of Cancer*, vol. 134, no. 3, pp. 542–551, 2014.