RESEARCH ARTICLE

Efficient $\ell_0$-norm feature selection based on augmented and penalized minimization

Xiang Li$^1$ | Shanghong Xie$^2$ | Donglin Zeng$^3$ | Yuanjia Wang$^2$

$^1$Statistics and Decision Sciences, Janssen Research & Development, LLC, Raritan, NJ, USA
$^2$Department of Biostatistics, Mailman School of Public Health, Columbia University, New York, NY 10032, USA
$^3$Department of Biostatistics, University of North Carolina, Chapel Hill, North Carolina 27599, USA

Correspondence
Yuanjia Wang, Department of Biostatistics, Mailman School of Public Health, Columbia University, New York, NY 10032, USA.
Email: yuanjia.wang@columbia.edu

Funding information
NIH, Grant/Award Number: NS073671, NS082062, CA082659 and GM047845
NINDS, Grant/Award Number: NS082062 and NS073671

Advances in high-throughput technologies in genomics and imaging yield unprecedentedly large numbers of prognostic biomarkers. To accommodate the scale of biomarkers and study their association with disease outcomes, penalized regression is often used to identify important biomarkers. The ideal variable selection procedure would search for the best subset of predictors, which is equivalent to imposing an $\ell_0$-penalty on the regression coefficients. Since this optimization is a nondeterministic polynomial-time hard (NP-hard) problem that does not scale with number of biomarkers, alternative methods mostly place smooth penalties on the regression parameters, which lead to computationally feasible optimization problems. However, empirical studies and theoretical analyses show that convex approximation of $\ell_0$-norm (e.g., $\ell_1$) does not outperform their $\ell_0$ counterpart. The progress for $\ell_0$-norm feature selection is relatively slower, where the main methods are greedy algorithms such as stepwise regression or orthogonal matching pursuit. Penalized regression based on regularizing $\ell_0$-norm remains much less explored in the literature. In this work, inspired by the recently popular augmenting and data splitting algorithms including alternating direction method of multipliers, we propose a 2-stage procedure for $\ell_0$-penalty variable selection, referred to as augmented penalized minimization-$L_0$ (APM-$L_0$). The APM-$L_0$ targets $\ell_0$-norm as closely as possible while keeping computation tractable, efficient, and simple, which is achieved by iterating between a convex regularized regression and a simple hard-thresholding estimation. The procedure can be viewed as arising from regularized optimization with truncated $\ell_1$ norm. Thus, we propose to treat regularization parameter and thresholding parameter as tuning parameters and select based on cross-validation. A 1-step coordinate descent algorithm is used in the first stage to significantly improve computational efficiency. Through extensive simulation studies and real data application, we demonstrate superior performance of the proposed method in terms of selection accuracy and computational speed as compared to existing methods. The proposed APM-$L_0$ procedure is implemented in the R-package APML0.

KEYWORDS
ADMM, biomarker signature, censored data, variable selection, $\ell_0$-penalty
Recent advances in high-throughput technologies in genomics and imaging yield unprecedentedly large numbers of prognostic biomarkers to be examined. The curse of dimensionality poses challenges for the traditional regression analysis when studying association between high-dimensional biomarkers and disease outcomes.\(^1\) To cope with the scale of the number of variables, many regularized methods, which introduce sparsity penalties to the regression models or likelihood functions, have been developed for simultaneous parameter estimation and variable selection.\(^2-7\) The most ideal penalty for the variable selection purpose is the \(\ell_0\)-norm of the regression coefficients for all predictors, which is equivalent to the number of nonzero terms among the coefficients, and also referred to as the best subset selection. Unfortunately, due to the nonconvexity and discontinuity of the \(\ell_0\)-norm, solving such a regularized optimization is computationally challenging, known as nondeterministic polynomial-time hard (NP-hard).\(^8\) Instead, other continuous or smooth penalties have been suggested in different contexts.\(^2-7\) Particularly, the convex penalty based on the \(\ell_1\)-norm,\(^2,3\) \(\ell_2\)-norm, or their combination\(^4,5\) was introduced as a relaxation of \(\ell_0\)-norm, providing a computationally attractive regularization form.

Alternative approaches based on nonconvex penalties such as smoothly clipped absolute deviation\(^6,10\) and approximate \(\ell_0\)-penalty\(^11,12\) apply less shrinkage on large coefficients and hence reduce the estimation bias. Moreover, nonconvex penalties may yield the property of oracle variable selection in the large sample sense. However, one difficulty of using nonconvex penalties is computational instability and sensitivity to initial values. None of these methods directly use the \(\ell_0\)-penalty and thus likely will still include some variables with small effects in the final model, especially under the high-dimensional data framework with large \(p\) and small \(n\). For example, Lin et al\(^13\) showed that \(\ell_1\)-regularized methods never outperform their \(\ell_0\)-counterpart and may be much worse in some cases. Advancement for \(\ell_0\)-norm feature selection is low, where the main methods are greedy algorithms such as stepwise regression or orthogonal matching pursuit.\(^14\)

Penalized regression based on regularizing \(\ell_0\)-norm remains much less explored in the literature. The penalty function proposed in Shen et al\(^15\) targets \(\ell_0\)-norm but involves heavy computation and nonconvex optimization.

To address gaps in knowledge, we propose an efficient 2-stage method that aims to regularize \(\ell_0\)-norm as close as possible and can be solved by a highly efficient and simple computational algorithm. Our method shares 2 features with the recently popular alternating direction method of multipliers (ADMM) algorithm\(^16\): (1) introducing surrogate parameters to augment the original model space; and (2) updating original parameters and surrogate parameters with iteratively alternating optimization. To describe the difference with the ADMM, note that it solves optimization problems of the form

\[
\min_{\beta, \theta} f(\beta) + g(\theta), \text{ subject to } A\beta + B\theta = C,
\]

where all \(f(\beta)\) and \(g(\theta)\) are convex functions. However, a fundamental difference is that \(g(\theta)\) is the \(\ell_0\)-norm of \(\theta\) in our method, so it is nonconvex. Using \(\ell_0\)-norm, our variable selection retains an authentic sparsity penalty. Another difference is that the ADMM obtains step sizes for parameter updates as solutions to the Lagrange equations. However, we can regard our soft-thresholding followed by hard-thresholding procedure as arising from a truncated \(\ell_1\)-penalty function and treat step sizes as tuning parameters. Thus, we will use cross-validation instead of Lagrange equations to determine their values, and our tuning parameters are chosen adaptively to the data at hand.

We refer to our method as the augmented penalized minimization-\(L_0\) (APM-\(L_0\)). Specifically, APM-\(L_0\) iterates between a commonly used regularized regression step and a hard-thresholding estimation step, which can avoid the computational challenges encountered in the \(\ell_0\)-regularization problems. To implement APM-\(L_0\), we develop a 1-step coordinate descent algorithm taking into account the sparsity structure, which results in both significant reduction in memory usage and high efficiency in computation. Furthermore, we propose to simultaneously tune the regularization parameters in both steps based on cross-validation. The method is flexible enough to handle a variety of models (eg, linear model, logistic model, or Cox proportional hazards model\(^17\)) and structure among variables by imposing a Laplacian penalty.\(^6,18\) We demonstrate better estimation accuracy, much improved model sparsity, and reduced computational burden over the commonly used \(\ell_1\)-type penalties via extensive simulation studies. We provide real data analyses to demonstrate the practical applicability of APM-\(L_0\). Lastly, a publicly available R-package APML0 is provided and shown via simulations to speed up computation faster than the commonly used R-package glmnet.\(^19\)

The rest of this article is organized as follows. In Section 2, we describe the \(\ell_0\)-penalized problems and present the details of APM-\(L_0\) approach. We also describe an efficient 1-step coordinate descent algorithm for the implementation. In Section 3, we first evaluate the estimation and selection performance of our method and show large efficiency gain in
simulation studies. In Section 4, we apply APM-L₀ to a real-world example: a recently completed comprehensive study on Huntington’s disease (HD), PREDICT-HD,²⁰ where the whole brain structural magnetic resonance imaging (MRI) measures are used to estimate a network regularized biomarker signature for the age-at-onset of HD. Lastly, we conclude with a few remarks in Section 5.

2 | METHODS AND COMPUTATIONAL ALGORITHM

2.1 | Regression model with ℓ₀-penalty

Let β denote a vector of coefficients in a regression model and let l(β) denote a log-likelihood function chosen appropriately depending on the outcome. For example, for continuous outcomes, l(β) is based on linear regression model; and for censored outcomes, l(β) is based on the partial likelihood under the Cox proportional hazards model¹⁷ (details in Section 2.5). With a large number of biomarkers including genomic and imaging features, directly maximizing l(β) may not be feasible and it is necessary to impose regularization and perform variable selection. The ideal but computationally infeasible feature selection is the best subset selection, that is, performing a regularized regression imposing penalty on the ℓ₀-norm of coefficients:

\[
\min_{\beta} -n^{-1}l(\beta) + \rho||\beta||_0,
\]

where ||β||₀ = \(\sum_{j=1}^{p} I(\beta_j ≠ 0)\) and β_j is the jth component of β.

However, due to the nonconvexity of the ℓ₀-norm, it is difficult to solve (1) with the penalty function \(p(\beta) = \rho||\beta||_0\), computationally known as NP-hard: To select the best subset of nonzero coefficients, we need to evaluate all the possible combinatorial subsets, which grows exponentially with the number of covariates. Existing approaches based on more continuous penalty functions (eg, ℓ₁-, ℓ_q-norm instead of ℓ₀-norm) may often select many nonzero β's with small magnitude, which leads to a nonparsimonious model and inferior prediction on independent data due to overfitting, a common challenge for high-dimensional data analysis with large p and small n.

Remark 1. In some applications, components of biomarker variables X exhibit correlation structure (eg, correlated gene expressions or brain imaging region of interest (ROI) measures). Such correlation can be naturally described by a network structure through a Laplacian matrix L associated with the network graph. For example, Li and Li⁸ and Huang et al¹⁸ discussed incorporating such a Laplacian quadratic penalty \(β^T L β\) into the log-likelihood to perform network-regularized variable selection. This penalty encourages smoothness of the coefficients of predictors that are linked on the network. To accommodate network-informed penalty function, the first term in (1) can be expanded to \(-n^{-1}l(β) + \rho_2 β^T L β\) by taking into account such a network structure, where \(ρ_2\) is a tuning parameter for the Laplacian prior and is selected by cross-validation.

2.2 | APM-L₀ for the ℓ₀-penalized variable selection

Our proposed computational method to solve the NP-hard problem (1), APM-L₀, is motivated by a class of proximal methods performing augmentation and splitting, including the ADMM. In a nutshell, APM-L₀ is a 2-stage iterative procedure where the first stage solves a regularized regression with computationally tractable penalty function, and the second stage performs hard-thresholding. The procedure is simple and highly computationally efficient. To illustrate the method, we reformulate the objective function (1) by augmenting the ℓ₀-norm of β with a surrogate parameter θ and bound the difference by a smooth convex function which guarantees convergence in the proximal of β:

\[
- n^{-1}l(\beta) + \rho||\theta||_0 \quad \text{subject to} \quad \sum_{j=1}^{p} \phi_j(|β_j - θ_j|) \leq c,
\]

where \(\phi_j(x)\) is a convex function satisfying \(\phi_j(0) = 0\) and \(\phi_j(|x|) ≥ 0\) for \(x ≠ 0\), and \(c ≥ 0\) is a tuning parameter. A common choice for \(\phi_j(|·|)\) is the ℓ₂-norm, where \(\phi_j(|x|) = x^2, j = 1, \ldots, p\). Denote by λ a penalty parameter (λ > 0). The Lagrangian form for (2) becomes

\[
L_λ(β, θ) = - n^{-1}l(\beta) + \rho||θ||_0 + λ \sum_{j=1}^{p} \phi_j(|β_j - θ_j|).
\]
To minimize (3) for a given \(\lambda\), APM-L\(_0\) iteratively update all parameters using the following algorithm: At the \(k\)th iteration,

\[
\hat{\beta}^{k+1} = \arg \min_{\beta} -n^{-1}l(\beta) + \lambda \sum_{j=1}^{p} \phi_j \left( |\beta_j - \hat{\beta}^k_j| \right),
\]

\[
\theta^{k+1} = \arg \min_{\theta} \rho||\theta||_0 + \lambda \sum_{j=1}^{p} \phi_j \left( |\theta_j^{k+1} - \theta_j^k| \right),
\]

where the superscript is the iteration counter. The algorithm is iterated until convergence.

Note that the above update Equation (4) is similar to updating a regularized regression. For (5), it is clear that minimization is performed component-wise. Hence, if \(\beta_j^{k+1} = 0\), then \(\theta_j^{k+1} = 0\); otherwise, the optimal solution is either \(\theta_j^{k+1} = \beta_j^{k+1}\) or \(\theta_j^{k+1} = 0\) depending on whether \(\phi_j(|\beta_j^{k+1}|)\) is larger than \(\rho/\lambda\). That is, for \(j = 1, \ldots, p\),

\[
\theta_j^{k+1} = \beta_j^{k+1} 1 \left( \phi_j \left( |\beta_j^{k+1}| \right) > \rho/\lambda \right).
\]

It can be seen from (6) that the \(\ell_0\)-penalty works as hard-thresholding the estimates obtained from the regularized regression in the first step in (4). Many convex penalties proposed in the literature are good choices of \(\phi_j(\cdot)\), including \(\ell_1\)-penalty,\(^{23}\) elastic net (a combination of \(\ell_1\)- and \(\ell_2\)-penalty),\(^{4,21}\) group Lasso\(^{22}\) and sparse group Lasso.\(^{28}\)

In summary, the APM-L\(_0\) approximates solutions to (1) via an ADMM-inspired iterative 2-stage method. The first stage replaces \(\ell_0\) penalty with another penalty function that provides computationally tractable optimization, and the second stage corresponds to hard-thresholding. From another view, the APM-L\(_0\) performs best subset selection based on the magnitude of \(\beta\) estimated from a regularized regression. By making use of the order and magnitude of \(\beta\), one can greatly reduce the computing time to evaluate \(\ell_0\)-norm.

### 2.3 Efficient computation in the first stage

When there is no closed form solution for the score function of a penalized regression (eg, in a Cox proportional hazards regression), we can apply a quadratic approximation\(^3\) at some point of the current estimate of \(\beta\) (details in Section 2.5). Previous algorithms such as Shen et al\(^{19}\) cyclically updated \(\hat{\beta}_j\), \(j = 1, \ldots, p\), until some convergence criterion was met at the local point \(\hat{\beta}\). Here, instead of Shen et al.,\(^{19}\) we take a 1-step coordinate descent approach to update \(\hat{\beta}_j\) only. Our 1-step algorithm substantially improves computational efficiency. For the \(j\)th component of \(\hat{\beta}\), we solve

\[
-n^{-1} \frac{\partial l(\beta|\hat{\beta}_{-j})}{\beta_j} + \lambda \frac{\partial \phi_j(\beta_j)}{\partial \beta_j} = 0,
\]

where \(l(\beta|\hat{\beta}_{-j})\) is the log-likelihood function with all the components fixed except the \(j\)th component, \(\hat{\beta}_j\). Furthermore, we construct an active set \(A = \{j : \hat{\beta}_j \neq 0\}\) at the outset and update those \(\hat{\beta}_j\) for \(j \in A\) only, which is efficient for handling sparse \(\beta\) by reducing the number of updates. Let \(\eta = (\eta_1, \ldots, \eta_n)^T = (\hat{\beta}^T X_1, \ldots, \hat{\beta}^T X_n)^T = X \hat{\beta}\). The APM-L\(_0\) algorithm is

1. (Initialization) Set \(\hat{\beta} = \hat{\eta} = 0\) and \(A = \emptyset\).
2. (Active set) Update \(A\) at \(\hat{\beta}\).
3. (Loop) Iterate until convergence of \(\hat{\beta}\): cyclically update \(\hat{\beta}_j\) by (7) for \(j \in A\).
4. Converge if no update of \(A\); otherwise, go to step 3 with the updated \(A\).

When evaluating for a path of \(\lambda\), we use the previous estimate \(\hat{\beta}\) and active set \(A\) as a warm start for the next \(\lambda\) and follow steps 2 to 4.

**Remark** 2. For a network graph-constrained log-likelihood, the Laplacian matrix \(L\) is often sparse. Hence, in the implementation, we use a sparse matrix to represent \(L\), which greatly reduces memory usage and enhances computational efficiency.

### 2.4 Simultaneous selection of tuning parameters

An intuitive understanding of APM-L\(_0\) is to iteratively perform (a) regularized regression as laid out in the previous section, and (b) perform hard-thresholding. The tuning parameter \(\lambda\) controls the degree of regularization for the estimators, and the ratio between \(\rho\) and \(\lambda\) determines the number of nonnull coefficients (sparsity of the model).
In some cases, the ADMM algorithm can be slow to converge. The original ADMM iteratively updates the Lagrangian multiplier \( \lambda \) as, at the \( k \)th iteration,

\[
\lambda^{k+1} = \lambda^k + \alpha^k \sum_{j=1}^{p} \phi_j(|\beta_j^k - \theta_j^{k+1}|),
\]

where \( \alpha^k > 0 \) is a step size. The \( \alpha^k \) needs to be appropriately chosen to ensure the dual function, defined as \( g(\lambda) = \inf_{\beta, \theta} L_\lambda(\beta, \theta) \), is increasing. Based on the dual optimal \( \lambda^* \) obtained by maximizing \( g(\lambda) \), we can recover the primal optimal estimates \( \beta^* \) and \( \theta^* \).

Instead of iteratively updating \( \beta, \theta, \) and \( \lambda \), we propose to treat \( \lambda \) as a tuning parameter and search over a set of grid points of \( \lambda \). At each fixed value of \( \lambda \), we update \( \hat{\beta} \) and \( \hat{\theta} \) based on the optimization problems (4) and (5). To save computational time, we advocate one iteration update for \( \beta \) and \( \theta \), and declare \( \hat{\theta} \) as our final estimate. Hence, it is feasible that the algorithm to solve the \( \ell_0 \)-penalty runs as fast as other regularized regressions (eg, \( \ell_1 \)-penalized regression). Thus, given \( \lambda \) and \( \rho \), our method is implemented in a 2-stage fashion:

\[
\hat{\beta} = \arg \min_{\beta} \frac{1}{n} l(\beta) + \lambda \sum_{j=1}^{p} \phi_j(|\beta_j|),
\]

\[
\hat{\theta} = \arg \min_{\theta} \rho ||\theta||_0 + \lambda \sum_{j=1}^{p} \phi_j(|\hat{\beta}_j - \theta_j|).
\]

The path of \( \lambda \) can be set as in Friedman et al.\(^{19} \) In the second stage, we arrange \( |\hat{\beta}| \) in decreasing order and directly choose the number of nonnull coefficients in \( \hat{\beta} \) by keeping the \( \kappa \) largest coefficients of \( |\beta| \). We set the path of \( \kappa \) from zero to the total number of nonnull coefficients of \( \hat{\beta} \). We propose to use cross-validation to simultaneously select both parameters \( \kappa \) and \( \lambda \). For example, we suggest to use least squared error for linear regression and partial-likelihood for Cox model\(^{24} \) as cross-validation criteria.

### 2.5 Examples of regression models and penalty functions

We can use any log-concave function/model to replace \( l(\beta) \) in the proposed APM-\( L_0 \). Let \( \mathbf{X}_i = (X_{i1}, \cdots, X_{ip})^T \) denote a vector of covariates and \( y_i \) be the response variable. For linear regression with continuous outcome, we use

\[
l(\beta) = -\sum_{i=1}^{n} (y_i - \beta^T \mathbf{X}_i)^2.
\]

For time-to-event outcomes (eg, age-at-onset of a disease) subject to independent censoring, we consider Cox model. Let \( T_i \) be the time-to-event of interest and \( C_i \) be the censoring time. Denote by \( T_i = \min(T_i, C_i) \) the observed event time or censoring time and denote by \( \delta_i = I(T_i \leq C_i) \) the event indicator, where \( I(\cdot) \) is an indicator function. We use the following log-likelihood function:

\[
l(\beta) = \sum_{i=1}^{n} \delta_i \left\{ \beta^T \mathbf{X}_i - \log \left( \sum_{k \in R_i} \exp \left( \beta^T \mathbf{X}_k \right) \right) \right\},
\]

where \( R_i = \{ k : \tilde{T}_k \geq \tilde{T}_i \} \) denotes the risk set at time \( \tilde{T}_i \). Because there is no closed form when solving (7), we use the method proposed by Simon et al.\(^{5} \) We approximate \( l(\beta) \) based on its second-order Taylor series expansion centered at \( \hat{\beta} \) (estimated \( \beta \) from the previous iteration) and further approximate the Hessian matrix of \( l(\eta) \) by it’s diagonal matrix to obtain the following function

\[
l(\beta) = \frac{1}{2} \sum_{i=1}^{n} \gamma_i(\tilde{\eta}) \left( \tilde{\eta}_i + u_i(\tilde{\eta})/\gamma_i(\tilde{\eta}) - \beta^T \mathbf{X}_i \right)^2,
\]

where

\[
\begin{align*}
\gamma_i(\tilde{\eta}) &= \sum_{i \in C_i} \delta_i \left\{ \exp(\tilde{\eta}_i) \sum_{k \in R_i} \exp(\tilde{\eta}_k) - \left( \exp(\tilde{\eta}_i) \right)^2 \right\}, \\
u_i(\tilde{\eta}) &= \sum_{i \in C_i} \delta_i \left\{ \exp(\tilde{\eta}_i) \sum_{k \in R_i} \exp(\tilde{\eta}_k) - \left( \exp(\tilde{\eta}_i) \right)^2 \right\} / \gamma_i(\tilde{\eta}),
\end{align*}
\]
where \( C_j = \{ i : T_i \leq T_j \} \). With little effort, we can extend APM-L_0 to other models, such as general linear models. In the subsequent numeric studies, we focus on linear and Cox models.

To reduce the risk of overfitting and incorporate prior biological information, Laplacian penalty is often considered to regularize estimation.

### 3 | SIMULATION STUDIES

#### 3.1 | Simulation design and results

We conducted extensive simulations to evaluate the performance of APM-L_0. We chose \( \phi_j(|x|) \) to be the \( \ell_1 \)-type penalty function, which provides a computationally attractive form to reduce dimensionality. We compared the proposed method with the commonly used \( \ell_1 \)-type penalized regressions including Lasso, Enet, Net and its adaptive version, adaptive network penalty (ANet) for both linear regression, and Cox model. For simplicity, we refer to these methods as “\( \ell_1 \)-type,” to be distinguished from our proposed APM-L_0.

To mimic potential correlation between covariates in real-world applications, we constructed \( X \) in independent blocks and the nodes within each block were correlated with a correlation of 0.5. Each block consisted of 5 nodes and 15 nodes/covariates from 3 blocks had nonzero effects on the outcome. We fixed the number of covariates in each node at 5 but varied the total number of covariates, denoted by \( p \). We considered 2 types of outcome in the simulation, continuous and censored data. For linear regression, we generated \( Y_i = \sum_{j=1}^{15} \beta_j x_{ij} + \epsilon_i \), where \( \beta_j = (-1)^j \times 2 \exp(-j-1)/15 \) and \( \epsilon_i \sim N(0, 1) \). For Cox model, the underlying hazard function was given by \( \rho(t|x_i) = \rho_0(t) \exp \left( \sum_{j=1}^{15} \beta_j x_{ij} \right) \), where \( \rho_0(t) \) was specified by a Weibull distribution with shape parameter 5 and scale parameter 2 and \( \beta_j = (-1)^j \times 2 \exp(-j-1)/15 \). The censoring status was generated from a uniform distribution and randomly assigned to 30% subjects.

In the simulations, we considered sample size \( n = 200 \) with various numbers of covariates. For each simulated dataset, the searching path for \( \lambda \) has a length of 20. Ten-fold cross-validation was applied to choose the optimal tuning parameters. Simulations were repeated 100 times. To evaluate estimation performance, we computed the sum of squared errors (SSE) of the estimated parameters. We also calculated the number of true positive covariates (TP; number of nonnull variables correctly selected in the final model) and the number of false positive covariates (FP; number of null variables incorrectly selected in the final model) as measures of the variable selection performance. For the Cox model, we computed the out-sample concordance index (C-index) using 100 random partitions of data into a training set and testing set.

Table 1 summarizes these simulation results. It can be seen from Table 1 that APM-L_0 significantly outperforms the commonly used \( \ell_1 \)-type methods based on cross validation of the partial likelihood (L-CVpl) in terms of both estimation accuracy and selection performance for all cases. The improvement is substantial with smaller SSE, comparable TP, and much less FP for both linear regression and Cox regression. As the number of covariates increases, it becomes more difficult to pick true positive covariates and remove noise covariates. For the setting with small number of covariates where \( p = 50 \), both methods are able to select all the true positive covariates but APM-L_0 selects 20 times fewer FP than the \( \ell_1 \)-type methods. As \( p \) increases to 10 000, less TP are selected, yielding larger SSE. When \( p = 10000 \), the \( \ell_1 \)-type methods select slightly more TP than APM-L_0, but still many more FP, and hence a worse SSE. Comparing different choices of the penalty functions in the first stage \( \phi_j(\cdot) \) in APM-L_0, we find that ANet performs the best since it takes into account of the correlation structure among covariates and adjusts the signs of highly linked covariates. APM-L_0 with Anet penalty gives a higher C-index than CVpl when \( p = 1000 \) or \( p = 10000 \). The remaining 3 penalties have similar performance.

We also performed additional simulations with APM-L_0 under a fixed \( \lambda \) and number of nonnull variables, \( \kappa \). We fixed \( \lambda \) at 0.01, 0.05, and 0.1 and \( \kappa \) at 10, 20, and 30 and used a Lasso penalty as an example. When comparing APM-L_0 with the best scenario of fixing \( \lambda \) and \( \kappa \), the former yields a higher C-index, a lower number of false positives when \( p = 50 \); and gives a slightly lower C-index but less number of false positives when \( p = 1000 \) or \( p = 10000 \).

#### 3.2 | Running time

We compared the running time of our R-package APML0 implementing APM-L_0 with glmnet under the same parameter settings for various sample sizes and numbers of covariates. As glmnet can only handle Enet and Lasso, the comparison was only performed for these 2 penalties. To make the algorithms comparable, we generated the path of tuning parameter \( \lambda \) from glmnet and used the same path in the first stage of our method. All calculations were performed on an Intel Xeon 2.13 GHz processor.
TABLE 1  Comparison of estimation and selection performance for linear regression and Cox model based on the proposed APM-\(L_0\) and existing \(\ell_1\)-type methods for penalties ANet, Net, Enet, and Lasso with various numbers of covariates

<table>
<thead>
<tr>
<th></th>
<th>SSE\textsuperscript{a}</th>
<th>TP\textsuperscript{b}</th>
<th>FP\textsuperscript{c}</th>
<th>C-index\textsuperscript{d}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CVpl APM-(L_0) CVpl APM-(L_0)</td>
<td>CFpl APM-(L_0) CFpl APM-(L_0)</td>
<td>CFpl APM-(L_0) CFpl APM-(L_0)</td>
<td>CFpl APM-(L_0) CFpl APM-(L_0)</td>
</tr>
<tr>
<td>Linear regression</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(n = 200, p = 50)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANet</td>
<td>0.26 0.18 15.0 15.0 25.4 1.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net</td>
<td>0.38 0.25 15.0 15.0 19.4 1.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enet</td>
<td>0.37 0.25 15.0 15.0 18.8 1.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lasso</td>
<td>0.37 0.25 15.0 15.0 18.9 1.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(n = 200, p = 1000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANet</td>
<td>1.86 1.39 14.8 14.7 213.4 1.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net</td>
<td>2.49 1.90 15.0 14.7 125.3 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enet</td>
<td>2.41 1.86 15.0 14.6 125.5 0.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lasso</td>
<td>2.13 1.51 15.0 14.9 116.7 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(n = 200, p = 10000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANet</td>
<td>14.57 13.60 11.6 10.3 438.2 4.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net</td>
<td>16.85 15.92 9.6 7.5 95.5 6.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enet</td>
<td>16.29 15.80 9.8 7.5 102.3 7.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lasso</td>
<td>15.67 15.28 9.9 7.7 99.8 7.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cox model</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(n = 200, p = 50)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANet</td>
<td>1.36 0.55 15.0 15.0 27.5 1.6 0.912 0.915</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net</td>
<td>2.35 0.89 15.0 15.0 19.3 0.5 0.901 0.906</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enet</td>
<td>2.35 0.89 15.0 15.0 19.3 0.5 0.901 0.907</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lasso</td>
<td>2.29 0.88 15.0 15.0 18.6 0.6 0.902 0.907</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(n = 200, p = 1000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANet</td>
<td>13.28 6.90 14.0 13.5 158.9 0.8 0.800 0.844</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net</td>
<td>19.59 14.39 11.5 10.3 39.2 2.0 0.704 0.710</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enet</td>
<td>19.28 13.89 11.6 10.5 38.8 1.7 0.703 0.712</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lasso</td>
<td>19.01 13.11 11.7 10.8 37.5 1.5 0.700 0.709</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(n = 200, p = 10000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANet</td>
<td>23.48 17.15 8.8 9.3 124.8 1.8 0.669 0.725</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net</td>
<td>25.74 23.74 6.0 5.6 25.8 2.9 0.655 0.663</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enet</td>
<td>25.72 24.12 5.5 4.9 19.9 3.7 0.646 0.644</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lasso</td>
<td>25.68 24.25 5.4 4.4 17.3 2.9 0.641 0.641</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\textsuperscript{a}SSE: Sum of squared error; \\
\textsuperscript{b}TP: Number of true positive covariates; \\
\textsuperscript{c}FP: Number of false positive covariates; \\
\textsuperscript{d}C-index: Concordance index.

Table 2 shows the running time comparison between APML\(0\) and glmnet. Our implementation was called from R, and most intensive computation codes were written in C++ and integrated with R using R-package Rcpp.\textsuperscript{25} In Table 2, we observe that APML\(0\) and glmnet have similar running time for linear regression but APML\(0\) runs faster than glmnet for Cox regression. Similar algorithm was used by both packages for linear regression. For Cox model, a quadratic approximation is needed at a local point. APML\(0\) takes 1-step coordinate descent at the local point rather than full optimization as done by glmnet. Obtaining high precision of estimates for the intermediate steps is not necessary. Similar idea was adopted in Mittal et al.\textsuperscript{26} Additional simulations with different distribution of covariates, correlations among covariates, and comparison with ADMM are presented in the Appendix.

4  ANALYSIS OF REAL DATA

There is increasing evidence that brain imaging markers are important biomarkers for predicting diagnosis and progression of neurodegenerative disorders.\textsuperscript{20,27,28} Current work in the clinical literature mostly perform univariate analyses to assess association between individual variables and disease outcome. However, theoretical investigation\textsuperscript{2} and various empirical studies\textsuperscript{29} suggest that simultaneous approaches based on penalized regressions may avoid overfitting and
provide more power than massive univariate approaches or greedy-search based stepwise regressions. Here, we take a whole-brain approach to evaluate all regional imaging measures simultaneously in predicting age-at-onset (AAO) of Huntington’s disease (HD). Regional brain atrophy measures obtained from structural magnetic resonance imaging (MRI) have been suggested as one of the most robust imaging biomarkers for HD.\textsuperscript{30} We analyzed the data from the newly completed PREDICT-HD study\textsuperscript{20} to predict AAO of HD using whole brain subcortical volumetric measures obtained from structural MRI. The regional summary volumetric measures were created by a fully automated procedure and preprocessed using Freesurfer 5.2 (http://surfer.nmr.mgh.harvard.edu). Details on the imaging marker preprocessing have been reported previously.\textsuperscript{20} Our analysis consists of 840 subjects who were at genetic risk of HD (CAG repeats length $\geq 36$ at the huntingtin gene).\textsuperscript{31} The median follow up time was 3 years and 128 subjects developed HD during the study. In our analysis, there were 8 clinical variables (gender, education, baseline total motor score from the UHDRS, and cognitive and functioning measures) and 28 subcortical MRI imaging ROI biomarkers measured at the baseline visit. To account for correlation among imaging measures, elastic net (Enet) penalty and Laplacian penalty was used for the function $\ell$ and functioning measures) and 28 subcortical MRI imaging ROI biomarkers measured at the baseline visit. To account for correlation among imaging measures, elastic net (Enet) penalty and Laplacian penalty was used for the function $\ell$.

We used control subjects (no HD mutation, CAG repeats length $< 36$) in PREDICT-HD to estimate the correlation matrix used in the Laplacian penalty. All variables were standardized before fitting the model. The proposed APM-\textsubscript{L0}, augmented penalized minimization-\textsubscript{L0}, has higher C-index, lower Brier score and higher partial likelihood than under Lasso penalty. We further show the estimated standardized regression coefficients (effect sizes) in Figure 1 for all 4 procedures. Comparing APM-\textsubscript{L0} with L-CVpl, we see that the former removed several biomarkers with small effects (eg, Left Lateral Ventricle) which were clearly noise variables from a biological point of view, while strengthening effects from ROIs such as Putamen, Thalamus, and Pallidum. Comparing Lasso penalty with sign-adjusted ANet, we see that the former does not select ROIs shown to be highly predictive in prior literature such as left or right side of Putamen. In addition, ANet can select the linked biomarkers with opposite effects, such as left and right sides of hippocampus, which indicates the necessity of controlling for the direction of association of biomarkers.

The variables selected by APM-\textsubscript{L0} are consistent with previously identified in the literature from the PREDICT-HD study. However, previous literature did not take a multivariate approach so that the relative ranking of biomarkers’ ability to predict HD onset in a multivariate model is unknown. Based on their effect sizes as shown in Figure 1, the top ranking clinical variables include total motor score, Stroop inference score, Stroop word score, and the

### TABLE 2

Running time in seconds for R-packages APM-\textsubscript{L0} and \texttt{glmnet} for various sample sizes and number of covariates

<table>
<thead>
<tr>
<th>Number of Variables</th>
<th>Linear Regression</th>
<th>Cox Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enet</td>
<td>Lasso</td>
</tr>
<tr>
<td></td>
<td>APM-\textsubscript{L0}</td>
<td>\texttt{glmnet}</td>
</tr>
<tr>
<td></td>
<td>APM-\textsubscript{L0}</td>
<td>\texttt{glmnet}</td>
</tr>
<tr>
<td>$n = 200, p = 1000$</td>
<td>0.11</td>
<td>0.27</td>
</tr>
<tr>
<td>$n = 200, p = 10000$</td>
<td>0.70</td>
<td>0.85</td>
</tr>
<tr>
<td>$n = 5000, p = 1000$</td>
<td>2.79</td>
<td>3.00</td>
</tr>
<tr>
<td>$n = 50000, p = 10000$</td>
<td>10.95</td>
<td>10.35</td>
</tr>
</tbody>
</table>

Abbreviations: APM-\textsubscript{L0}, augmented penalized minimization-\textsubscript{L0}.

### TABLE 3

Average number of variables selected, C-index, Integrated Brier Score and Partial Likelihood by the proposed APM-\textsubscript{L0}, L-CVpl with ANet, Net, Enet and Lasso penalty (100 repetitions of 10-fold cross validation)

<table>
<thead>
<tr>
<th>Number of Variables</th>
<th>C-index</th>
<th>Integrated Brier Score</th>
<th>Partial Likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ANet</td>
<td>Lasso</td>
<td>ANet</td>
</tr>
<tr>
<td>L-CVpl</td>
<td>27.32</td>
<td>14.44</td>
<td>0.800</td>
</tr>
<tr>
<td>APM-\textsubscript{L0}</td>
<td>19.15</td>
<td>10.30</td>
<td>0.792</td>
</tr>
</tbody>
</table>

Abbreviations: APM-\textsubscript{L0}, augmented penalized minimization-\textsubscript{L0}; CVpl; cross validation of the partial likelihood.
top ranking imaging biomarkers include Thalamus, Putamen, Caudate, Hippocampus ROIs, and cerebellum white matter and cerebellum cortex. The symbol digital modality (SDMT) drops out of the model when Stroop scores are selected into the model. Noisy markers such as left and right lateral ventricle are not selected into the model.

Due to better interpretability and prediction performance, we present further results of APM-$L_0$ under ANet penalty. We estimated the structural covariation network\cite{10} from control subjects (no HD mutation) in PREDICT-HD. The estimated network was then used to construct Laplacian penalty in the estimation of the effects of ROIs. Thus, the highly correlated ROIs were encouraged to express similar effects as in Li and Li.\cite{11} We show in Figure 2 the imaging network signature and their effect sizes. For graphical presentation purpose, Figure 2 only displays estimated nonnull ROIs and their strongly associated edges. Each edge represents 2 ROIs with the absolute value of correlation greater than a threshold (0.8) in the covariation network and the size and color of nodes represents the effect and direction of an ROI on the age at onset of HD, respectively. More ROIs were chosen by L-CVpl compared with APM-$L_0$. The networks identified by L-CVpl with small effects were removed by APM-$L_0$ (ie, Choroid Left-Choroid Right). Furthermore, the effects of important ROIs

![FIGURE 1](image1.png)

**FIGURE 1** Forest plot of standardized effect sizes for biomarkers selected by CVpl Lasso, APM-$L_0$ Lasso, CVpl Anet, APM-$L_0$ Anet. APM-$L_0$, augmented penalized minimization-$L_0$; CVpl; cross validation of the partial likelihood

![FIGURE 2](image2.png)

**FIGURE 2** Comparison of network identified by ANet based on L-CVpl (left (A)) and proposed APM-$L_0$ (right (B)), with radius indicating effect sizes and color indicating signs of effects (blue: positive; red: negative). APM-$L_0$, augmented penalized minimization-$L_0$; CVpl; cross validation of the partial likelihood
were strengthened by APM-$L_{0}$, indicated by larger radius of nodes in Figure 2. These results show that APM-$L_{0}$ has the desirable properties of amplifying effect sizes of important ROIs and eliminating noisy ROIs.

To assess the ability of biomarkers in discriminating individuals who will have an onset of HD by certain age $t$ from those who will not, we split subjects into high-risk and low-risk groups based on their biomarker risk scores (ie, $\beta^T X$). Receiver operating characteristic curve (ROC) analysis is applied to select the optimal cutoff values of risk scores for predicting risk by age $t =$40, 50, 60, or 70, respectively. The cutoff values are determined by minimizing the difference between the points on the ROC curve and the point (0,1) on the upper left hand corner of ROC space. Subjects are divided into high-risk group and low-risk group based on the optimal cutoff values. Figure 3 shows the cumulative risk of developing HD in high-risk group and low-risk group estimated from Kaplan-Meier curves. It indicates a large difference between the high-risk group and low-risk group. We computed time-dependent AUCs using the method in Chiang and

**FIGURE 3** Estimated cumulative risk of Huntington disease diagnosis using augmented penalized minimization-$L_{0}$ Anet. From left to right, results are obtained at age 40, 50,60, and 70. Blue: high-risk group. Red: low-risk group

**FIGURE 4** Two-biomarker split plots using augmented penalized minimization-$L_{0}$ Anet. The top row shows Pallidum-Left versus Thalamus-Left. The bottom row shows Pallidum-Left versus Thalamus-Right. From left to right, the cutoff values are optimized for distinguishing onset by age 40, 50, 60, and 70. Blue: high-risk group. Red: low-risk group. Black line: separation boundary. Large filled circles: subjects with a diagnosis by certain age. Dots: subjects without a diagnosis by certain age
Hung\textsuperscript{36} that can account for censoring implemented in the R package “timeROC.” The AUCs of risk scores obtained from APM-L\textsubscript{0} is high: at age 40, 50, 60, or 70, the AUCs are 0.84, 0.87, 0.91, and 0.89, respectively. To visualize the ability of biomarkers with largest effect sizes in discriminating high- and low-risk individuals, we present 2-biomarker split plots in Figure 4. The decision boundary in each figure is obtained by fixing other biomarkers at the sample averages. They show some discriminant power for separating high-risk group and low-risk group by Pallidum-Left and Thalamus-Left, or Pallidum-Left and Thalamus-Right, especially at \( t = 50 \) or \( t = 60 \). For lower or higher age, the discriminant power of the two top ranking biomarkers is limited and borrowing information from other biomarkers is necessary to achieve higher predictive performance.

5 | DISCUSSION

In this work, we propose a 2-stage procedure under the ADMM framework to approximate solutions to the \( \ell_0 \)-penalty variable selection. We develop an efficient 1-step coordinate descent algorithm for implementation. Our APM-L\textsubscript{0} approach improves both estimation and selection performance substantially over the commonly used regularized methods. The 1-step coordinate descent algorithm runs faster than existing algorithms which fully optimize the estimates at each step. Taking into account the sparsity structure allows for further improvement on the computation efficiency.

Here, we focus on linear regression and Cox model, and demonstrate the procedure mainly using \( \ell_1 \)-type penalties in the first stage for \( \phi_j \). However, the proposed approach can easily be extended to other types of outcomes and penalty forms. One would replace the log-likelihood function with any other log-concave function to obtain a similar procedure. It would be interesting to explore other shrinkage methods, such as smoothly clipped absolute deviation\textsuperscript{10} or MCP.\textsuperscript{37} We expect similar results such that APM-\( \ell_0 \) would achieve better sparsity and accuracy than alternative methods. Furthermore, our algorithm can be improved and easily adjusted for massive sample-size data by accounting for the sparsity in the covariates matrix. Lastly, in this work, baseline biomarkers are used to predict disease onset. Another extension worth considering is the inclusion of longitudinal measures of biomarkers over time in a time-dependent model to update predictive function for disease onset.

ACKNOWLEDGEMENTS

This work is supported by NIH grants NS073671, NS082062, CA082659, GM047845. The authors wish to thank the NIH dbGap data repository (accession number phs000222.v3.p2) and the PREDICT-HD study investigators.

ORCID

Xiang Li \( \text{http://orcid.org/0000-0001-9790-5663} \)
Yuanjia Wang \( \text{http://orcid.org/0000-0002-1510-3315} \)

REFERENCES


APPENDIX A: R-PACKAGE

R-package **APML0** contains R codes to perform all the methods considered in the simulation, including penalties of Lasso, Enet, Net, and ANet for both linear regression and Cox model. Most intensive computation codes were written in C++ and integrated to R codes using R-package **Rcpp**. R-package **APML0** is available upon request, and will be uploaded to CRAN.
TABLE B1  Comparison of the estimation and selection performance for Cox model based on APM-L₀ and existing ℓ₁-type methods for penalties ANet, Net, Enet and Lasso when covariates have different distributions

<table>
<thead>
<tr>
<th></th>
<th>SSE^a</th>
<th>C-index^b</th>
<th>TP^c</th>
<th>FP^d</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CVpl</td>
<td>APM-L₀</td>
<td>CVpl</td>
<td>APM-L₀</td>
</tr>
<tr>
<td>n = 200, p = 50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANet</td>
<td>2.01</td>
<td>0.90</td>
<td>0.915</td>
<td>0.918</td>
</tr>
<tr>
<td>Net</td>
<td>2.74</td>
<td>0.91</td>
<td>0.904</td>
<td>0.909</td>
</tr>
<tr>
<td>Enet</td>
<td>2.75</td>
<td>0.93</td>
<td>0.904</td>
<td>0.909</td>
</tr>
<tr>
<td>Lasso</td>
<td>2.89</td>
<td>1.22</td>
<td>0.905</td>
<td>0.909</td>
</tr>
<tr>
<td>n = 200, p = 1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANet</td>
<td>10.56</td>
<td>3.42</td>
<td>0.826</td>
<td>0.859</td>
</tr>
<tr>
<td>Net</td>
<td>20.73</td>
<td>13.64</td>
<td>0.710</td>
<td>0.727</td>
</tr>
<tr>
<td>Enet</td>
<td>20.46</td>
<td>12.37</td>
<td>0.710</td>
<td>0.730</td>
</tr>
<tr>
<td>Lasso</td>
<td>19.49</td>
<td>11.50</td>
<td>0.709</td>
<td>0.730</td>
</tr>
<tr>
<td>n = 200, p = 10000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANet</td>
<td>24.46</td>
<td>15.09</td>
<td>0.657</td>
<td>0.733</td>
</tr>
<tr>
<td>Net</td>
<td>26.48</td>
<td>24.95</td>
<td>0.645</td>
<td>0.659</td>
</tr>
<tr>
<td>Enet</td>
<td>26.50</td>
<td>25.34</td>
<td>0.639</td>
<td>0.642</td>
</tr>
<tr>
<td>Lasso</td>
<td>26.54</td>
<td>25.81</td>
<td>0.634</td>
<td>0.631</td>
</tr>
</tbody>
</table>

^aSSE: Sum of squared error;  
^bC-index: Concordance index;  
^cTP: Number of true positive covariates;  
^dFP: Number of false positive covariates.

APPENDIX B: ADDITIONAL SIMULATION STUDIES TO COMPARE WITH ADMM

In additional simulation studies, we varied the distributions of covariates. We constructed X in independent blocks, where each block consisted of 5 covariates and 15 covariates from 3 blocks had nonzero effects on the outcome. Covariates from the first nonzero effect block followed standard normal distribution and the covariates within the block were correlated with a correlation of 0.5. In the second block, covariates followed normal distribution with mean 0 and variance 1.5, and the variables within the block were correlated with a correlation of 0.5. In the third block, covariates followed noncentral t-distribution with noncentrality parameter 2 and degree of freedom 4, and the variables within the block were independent with each other. We considered sample size n = 200 with various numbers of covariates.

Table B1 summarizes these simulation results. It can be seen that APM-L₀ much outperforms the commonly used ℓ₁-type methods based on cross validation of the partial likelihood (L-CVpl) in terms of both estimation accuracy and selection performance for all cases when covariates have different distributions. The improvement is substantial with a smaller SSE, comparable TP, much less FP and a higher C-index (especially when p = 10000 with Anet penalty). ANet still performs the best similar to the scenario when the distribution of covariates is the same.

The APM-L₀ uses surrogate parameters similar to the proximal splitting based algorithms (ADMM algorithm is a special case). To see the difference with ADMM, first note that ADMM implemented in Boyd et al. optimizes the following

\[-n^{-1}l(\beta) + \rho||\theta||_1, \text{ subject to } \sum_{j=1}^{p}(\beta_j - \theta_j)^2 \leq c.\]  

(B1)

However, APM-L₀ replaces the ℓ₁-norm of \( \theta \) in the above objective function by an ℓ₀-penalty and uses a general sparsity inducing penalty function \( \phi_\ell(\cdot) \) to bound the difference between \( \theta \) and \( \beta \) instead of restricting to a quadratic function (see also (2). There is no existing literature on using ADMM to handle ℓ₀-norm. For implementation, APM-L₀ transforms the constrained form (2) to its Lagrange form (3) as

\[(\beta, \theta) = \arg \min_{\beta, \theta} -n^{-1}l(\beta) + \rho||\theta||_0 + \lambda \sum_{j=1}^{p} \phi_\ell(|\beta_j - \theta_j|),\]  

(B2)

and simultaneously selects tuning parameters \((\lambda, \rho)\) based on cross-validation. In contrast, ADMM determines the step sizes of update functions (the equivalence of tuning parameters) by directly solving Lagrange equations instead of choosing them in a data-adaptive fashion from cross-validation.
TABLE B2  Estimation and selection performance of ADMM with fixed \( \lambda \) for linear regression

<table>
<thead>
<tr>
<th>( n = 200, p = 50 )</th>
<th>( n = 200, p = 1000 )</th>
<th>( n = 200, p = 10000 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSE(^a)</td>
<td>TP(^b)</td>
<td>FP(^c)</td>
</tr>
<tr>
<td>0.01</td>
<td>0.54</td>
<td>15.00</td>
</tr>
<tr>
<td>0.05</td>
<td>0.55</td>
<td>15.00</td>
</tr>
<tr>
<td>0.10</td>
<td>0.55</td>
<td>15.00</td>
</tr>
<tr>
<td>0.50</td>
<td>0.51</td>
<td>15.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.43</td>
<td>15.00</td>
</tr>
<tr>
<td>5.00</td>
<td>0.33</td>
<td>15.00</td>
</tr>
<tr>
<td>10.00</td>
<td>0.40</td>
<td>15.00</td>
</tr>
<tr>
<td>20.00</td>
<td>0.97</td>
<td>15.00</td>
</tr>
</tbody>
</table>

\(^a\)SSE: Sum of squared error; \(^b\)TP: Number of true positive covariates; \(^c\)FP: Number of false positive covariates.

Since ADMM uses \( \ell_1 \)-penalty for \( \theta \), we compared it to APM-\( L_0 \) with Lasso penalty. The simulation settings are as the same as in Section 4. We evaluated different values of the tuning parameter given at 0.01, 0.05, 0.1, 0.5, 1.0, 5.0, 10.0, and 20.0. Table B2 summarizes the results of ADMM under a linear regression model. Comparing to results of APM-\( L_0 \) in Table 1, we can see that APM-\( L_0 \) has a smaller SSE, comparable TP and much smaller FP. For \( p = 50 \), both ADMM and APM-\( L_0 \) can correctly choose all true covariates, but APM-\( L_0 \) selected more than 20 times fewer FP than ADMM. When \( p = 10000 \), ADMM selected more TP variables, but at the price of many more FPs. The number of iterations required for ADMM to converge can be more than 1000 and thus the computational speed is much slower than APM-\( L_0 \) in some scenarios.