The Application of Alternating Direction Method of Multipliers on $l_1$-norms Problems

Yanchen He$^1$ $^{a}$

$^1$School of statistics and mathematics, Central University of Finance and Economics

$^a$ Yanchen He: yanchenhe@foxmail.com

Abstract. $l_1$ regularization is commonly used for variable or model selection, and it is often incorporated into the penalized likelihood framework for the sparse solutions. Lasso, group lasso and graphical lasso are three representative methods among them. The objective functions of these methods all contain a $l_1$ penalized term. Several algorithms have been developed to solve those problems, such as least angle regression for lasso, group nonnegative garrote for group lasso and block coordinate descent for graphical lasso. However, they are designed for the solution of a specific problem and none of them can solve $l_1$-norm problems generally. A convex optimization algorithm called ADMM(Alternating Direction method of multipliers)splits targets problems into two distinct parts and handles them separately, which makes it a natural fit for $l_1$-norm problems consisting of a likelihood term and a penalty term.

Based on the methodology of ADMM, we derived a framework for the solution of lasso, group lasso and graphical lasso. The proposed framework outperforms the existing methods in simulations. Finally, we apply our method to analyze a birth weight dataset birthwt which could be found in R package MASS.

1. Introduction
We are now embracing an era of big data. With the rapid growth of the data generated from internet statistical models tend to be more and more complex. Such complexity could be caused by, for example, the huge increase of the explanatory variables for a response variable. However sometimes people only seek for these variables which are “the most relevant to the response variable” in a statistical model. Here are some reasons for that:

1: One may seek for simpler way to explain the data.
2: One may try to reduce the complexity of model by selecting a reasonable subset of explanatory variables because of the need to reduce the cost of prediction.
3: Redundant explanatory variables may cause colinearity. To avoid colinearity one could apply variable selection.

In practice there are a cluster of methods for model selection. Among them a popular approach is to add a $l_1$ penalty term to the model. The geometric property of $l_1$ penalty ensures its better performance in variable selection since it enforces the solution to be sparse (For example, $l_1$ penalty performs better than $l_2$ penalty because of their different constraints boundaries[1]. The shape of the boundaries of $l_1$ penalty makes some variables shrink to 0 exactly.), thus $l_1$ penalty is applied to a wide range of statistical model like LASSO[1] for variable selection, group lasso for grouped variables selection[2], graphical lasso for determining the direct interaction in Gaussian graph problem[3].
The LASSO(Least Absolute Shrinkage and Selection Operator), proposed firstly by R.Tibshirani in 1996 [1], adds a $l_1$ penalty term to the least square loss in order to select the key factors and increase the prediction accuracy. Let $A$ and $b$ be the design matrix of explanatory variables and vector of observations respectively, and denote $x$ the vector of coefficients. The aim of LASSO is to minimize:

$$\frac{1}{2} \|Ax-b\|^2 + \lambda \|x\|_1$$

(1.1)

where $\lambda$ is the tuning parameter. To solve the LASSO problem, several methods have been proposed. B.Efron et al. developed LARS(Least-Angle Regression) [4], which gets the solutions by adjusting the coefficients in the direction that make all the correlations with the residual the same. The coordinate descent method, which in each step minimizes the function in coordinate direction, is also applicable to LASSO. Early introductions of such application can be seen on Fu[5].

Group lasso, introduced by M.Yuan and Y.Lin[3], was motivated by the idea that in a linear model one may have several groups of variables but only a few groups of them are useful. Let the coefficients of vector be grouped as follows:

$$x = \begin{pmatrix} x_1^T & x_2^T & \ldots & x_T^T \end{pmatrix}^T$$

The group lasso problem is to minimize the following objective function:

$$\frac{1}{2} \|Ax-b\|^2 + \lambda \sum_{i=1}^{T} \|x_i\|_1$$

(1.2)

In the paper of M.Yuan and Y.Lin, they developed Group LARS for solving group lasso problem, which is a generalized version of LARS. They also extended the coordinate descent method to block coordinate descent method, in which every step focuses on optimizing the objective function with respect to one group instead of one coordinate.

In Gaussian graphical model, for a dataset generated from a multivariate Gaussian distribution with mean $\mu$ and covariance $\Sigma$ (the covariance is assumed to be positive definite so that the precision matrix $X=\Sigma^{-1}$ is well-defined), it has been proved that (see, for example, [6]) two components $x_i$ and $x_j$ of the random variable are conditional independent if and only if $X_{ij} = X_{ji} = 0$, thus estimating the inverse matrix of the covariance matrix $\Sigma^{-1}$ is crucial in the detection of relation between factors in the model. Banerjee[7], Yuan and Lin[8] proposed graphical lasso, a $l_1$-penalized model to solve the estimating problem. Given the empirical covariance matrix $S$, the estimation problem requires to minimize:

$$\text{tr}(SX) - \log \det(S) + \lambda \|X\|_1$$

(1.3)

where $S$ and $X$ are symmetric positive definite matrices and $\|\|$ is elementwise $l_1$ norm, that is, $\|X\|_1 = \sum_{i} |X_{ij}|$.

For the algorithm solving graphical lasso, in the original paper of Yuan and Lin they solve it using interior point algorithm. Jerome H.Friedman, Trevor Hastie and Robert Tibshirani handle the problem by repeatedly solving LASSO to determine a sub-block of the matrix.

Although some algorithms have been designed to deal with the problems above, it should be noted that all of them are designed to solve some specific problems. It’s natural for us to seek for an algorithm that can deal with all of the problems we mentioned, or, to speak more generally, all the optimization problems with $l_1$ penalty.

Alternative Direction Method of Multiplier (ADMM), originally proposed by Glowinski and Gabay[9] and reviewed by Boyd et al.[10], is an algorithm suitable for decomposable problems. For a constrained optimization with a decomposable objective function, ADMM separate the problem into several independent parts (here the term “independent” means each part relies on only one coordinate) and deal with each one separately. Such feature of ADMM indicates that it is suitable for $l_1$ penalized problem since $l_1$-penalized problem consists of a likelihood part and a $l_1$ penalty part, and by introducing new variable we can make the problem decomposable. For specific problems, especially $l_1$
-penalized problem, ADMM has been proved to be efficient with high convergence speed[10]. Due to ADMM’s good performance in solving $l_1$-penalized problem, in this article, we use ADMM algorithm to solve the three typical $l_1$-penalized problem: LASSO, Group lasso and graphical lasso.

The rest of the article is organized as follows. In section 3 we introduce the general procedure of ADMM algorithm and the background of three classical $l_1$-penalized problems, including lasso, group lasso and graphical lasso. The framework for the solution of these problems is derived. In section 4, several simulations are conducted and we compare the accuracy and efficiency of our proposed method with other existing algorithms. Finally in section 5, we demonstrate our methods by an application to the birth weight data served in R package MASS.

2. ADMM approach to solve $l_1$ problem

2.1. General framework of ADMM

ADMM solves problems in the form:

$$\min f(x) + g(z)$$

s.t. $Ax + Bz = c$ (2.1)

where $f$ and $g$ are convex functions.

Firstly, we write the augmented Lagrange function of the objective function as

$$L_p(x, z, y) = f(x) + g(z) + y^T (Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|^2$$ (2.2)

Let $u = (1/\rho)y$, and we may simplify it as:

$$L_p(x, z, y) = f(x) + g(z) + \frac{\rho}{2} \|Ax + Bz - c + u\|^2 - \frac{\rho}{2} \|u\|^2$$ (2.3)

ADMM repeats the following steps until convergence:

$$x^{k+1} = \arg \min_x \left(f(x) + \frac{\rho}{2} \|Ax + Bz^k - c + u^k\|^2\right)$$ (2.4)

$$z^{k+1} = \arg \min_z \left(f(x) + \frac{\rho}{2} \|Ax^{k+1} + Bz - c + u^k\|^2\right)$$ (2.5)

$$u^{k+1} = u^k + Ax^{k+1} + Bz^{k+1} - c$$ (2.6)

In (2.4), we first solve for $x$ that minimizes the objective function with $z$ and $u$ fixed, then we look for $z$ that minimize with $x$ and $u$ fixed in (2.5), and finally the dual variable $u$ is updated.

Now we try to solve $l_1$ penalized problem via ADMM. Problems with $l_1$ regularization often take the below form:

$$\min \ell(x) + \lambda \|x\|_1$$ (2.7)

where $\ell(x)$ is a convex loss function. To make the application of ADMM possible, we shall state problem (2.7) in another way. By introducing a new variable $z$ we rewrite the problem in an equivalent form

$$\min \ell(x) + \lambda \|x\|_1$$

s.t. $x = z$ (2.8)

The ADMM algorithm for $l_1$ problem is

$$x^{k+1} = \arg \min_x \left(\ell(x) + \frac{\rho}{2} \|x - z^k + u^k\|^2\right)$$ (2.9)

$$z^{k+1} = S_{\lambda/\rho}\left(x^{k+1} + u^k\right)$$ (2.10)

$$u^{k+1} = u^k + x^{k+1} - z^{k+1}$$ (2.11)

where $S_{\lambda}(x) = \left(1 - \frac{\lambda}{\|x\|^2}\right)x$ is the soft thresholding operator.
2.2. Three specific problems

2.2.1. LASSO

In 1996, Tibshirani proposed a new method, called LASSO, to solve linear regression problem. He added a $l_1$ penalty to the sum of squared residuals[1]. The benefit brought by $l_1$ penalty is that the geometric property of $l_1$ norm ensures that estimated values of some parameters can be shrunk to 0 precisely, thus increase the accuracy and interpretability of the linear model. LASSO has a wide range of applications. For example, in video concept detection, an adjusted form of LASSO(parallel lasso) was proposed to build the detectors of visual features [11].

Recall that the LASSO problem is to minimize

$$\frac{1}{2} \|Ax-b\|_2^2 + \lambda \sum |x_i|$$  

(1.1)

Applying the ADMM procedure (2.9), (2.10) and (2.11) for $l_1$ problem, we obtain following algorithm:

$$x^{k+1} = (A^TA + \rho I)^{-1} (A^Tb + \rho (x^k - u^k))$$  

(2.12)

$$z^{k+1} = S_{\lambda, \rho} (x^{k+1} + u^k)$$  

(2.13)

$$u^{k+1} = u^k + x^{k+1} - z^{k+1}$$  

(2.14)

The updating step for $x$ is obtained by replacing the loss-function term in (2.9) with $\frac{1}{2} \|Ax-b\|_2^2$, compute the gradient of (2.9) and let the gradient be 0.

2.2.2. Group lasso

The group lasso, firstly proposed by Ming Yuan and Yi Lin[2], stresses the group structure in parameters. It replaces the penalty term inside classical LASSO with groupwise $l_2$ penalties. The group structure occurs in many circumstances. For example, in model of categorical variables, a factor with several levels could be represented by several dummy variables. And it’s reasonable that these dummy variables should be considered together.

Recall that the group lasso problem is to minimize

$$\frac{1}{2} \|Ax-b\|_2^2 + \lambda \sum_i |x_i|$$  

(2.15)

where $x = (x_1^T, x_2^T, ..., x_N^T)^T$

This is not a classical $l_1$ problem. However ADMM approach is still applicable. Rewrite this problem:

$$\min \frac{1}{2} \|Ax-b\|_2^2 + \lambda \sum_i |z_i|$$  

s.t $z = (z_1^T, z_2^T, ..., z_N^T)^T$  

(2.16)

Now form the augmented Lagrangian with the scaled dual variable like what we do in (3.3):

$$\mathcal{L}(x,z,y) = \frac{1}{2} \|Ax-b\|_2^2 + \lambda \sum_i |z_i| + \frac{\rho}{2} \|x-z+u\|_2^2 - \frac{\rho}{2} \|y\|_2^2$$  

(2.17)

The step of updating $x$ is the same as the case of LASSO since they share the same parts involving $x$. For the step of updating $z$, we first group $z$, $c$ and $u$ in the same way as $x$ :

$$z = (z_1^T, z_2^T, ..., z_N^T)^T, c = (c_1^T, c_2^T, ..., c_N^T)^T, u = (u_1^T, u_2^T, ..., u_N^T)^T$$

Then it is easy to obtain the following equation:

$$\frac{\rho}{2} \|x-z+u\|_2^2 = \sum_i \frac{\rho}{2} \|z_i - u_i\|_2$$  

(2.18)
It means that we can update each $z_i$ separately. Plug (2.17) into (2.18) we get:

$$L_x(x, z, y) = \frac{1}{2} \| x - t \|_2^2 + \sum_{i} \left( \lambda_i \| z_i \|_2 + \frac{\rho_i}{2} \| z_i^* - z_i + u_i \|_2^2 \right) - \frac{\rho_i}{2} \| u_i \|_2^2 \tag{2.19}$$

So for each $z_i$ the task is to find $z_i$ that minimize:

$$\lambda \| z_i \|_2 + \frac{\rho_i}{2} \| z_i^* - z_i + u_i \|_2^2 \tag{2.20}$$

The solution to that problem is:

$$z_i^{k+1} = S_{ij}' \left( z_i^{k+1} + u_i^k \right) \tag{2.21}$$

Now we can summarize the 3 steps in one iteration:

$$x^{k+1} = (A^T A + \rho I)^{-1} \left( A^T b + \rho \left( z^k - u^k \right) \right) \tag{2.22}$$

$$z_i^{k+1} = S_{ij}' \left( z_i^{k+1} + u_i^k \right) \text{ for each sub-vector } z_i \tag{2.23}$$

$$u^{k+1} = u^k + x^{k+1} - z^{k+1} \tag{2.24}$$

### 2.2.3. Graphical lasso

As stated before, in the Gaussian graphical model determining the sparse pattern of $\Sigma^{-1}$ is important since $(\Sigma^{-1})_{ij} = 0 \text{ if and only if the } i\text{-th component and } j\text{-th component of the variable are not connected}$. To estimate $\Sigma^{-1}$ we often minimize the negative likelihood of the distribution. Such problem is called sparse inverse covariance estimation. A typical way to solve the problem is to apply graphical lasso, which add a $1_1$ penalty to the negative log-likelihood of $\Sigma^{-1}$. Recall that the sparse inverse covariance estimation problem is to minimize

$$\text{tr}(SX) - \log \det(X) + \lambda \|X\|_1 \tag{2.25}$$

where $S$ and $X$ are symmetric positive definite matrices and $\|\cdot\|_1$ is elementwise $l_1$ norm, namely,

$$\|X\|_1 = \sum_{i,j} |X_{ij}|.$$

Convert it into ADMM form:

$$\min_{X} \text{tr}(SX) - \log \det(X) + \lambda \|X\|_1 \tag{2.26}$$

s.t. $X = Z$

Now deduce the augmented Lagrangian form of (2.26)

$$L_x(x, z, y) = \text{tr}(SX) - \log \det(X) + \lambda \|X\|_1 + \rho \|X - Z\|_2^2 \tag{2.27}$$

The updating step for $X$ is computed by taking the gradient of $X$ and force it to vanish.

$$X^{k+1} = QYQ^T \tag{2.28}$$

where $Q$ is obtained by spectral decomposition:

$$\rho \left( Z^k - U^k \right) - S = Q \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & \lambda_s \end{bmatrix} Q^T$$

$X$ is a diagonal matrix and $Y = \frac{\lambda_i + \sqrt{\lambda_i^2 + 4\rho}}{2\rho}$.

The updating step for $Z$ is elementwise optimization:

$$Z_i^{k+1} = S_{ij}' \left( X_i^{k+1} + U_i^k \right) \tag{2.29}$$
3. Simulation

In this part we simulate some examples related to the three models (LASSO, group lasso and graphical lasso) and apply our proposed methods to solve them. The purpose is to evaluate the efficiency and accuracy of three ADMM algorithms proposed in the last section. We compare the efficiency of ADMM with other commonly used methods such as coordinate descent (CD) and block coordinate gradient descent (BCGD)[13]. When comparing the time cost, we run each algorithm for 100 times and compute the average of the total time for each method. The results are shown roughly in figures and exact numerical results are in appendix. In all the figures in the rest of the article the black bars indicate the time cost of ADMM while the white bars indicate the time cost of function to compare with. For the test of numerical accuracy we first compute the difference between ADMM’s result and the result of other commonly used algorithm corresponded to the problem, and then check the $l_1$ norm of the difference. The norm will be listed in a series of tables.

All the observations represented by the rows of the matrix $X$ are generated from a multivariate Gaussian distribution $N(0,\Sigma)$. Here we consider two cases: The explanatory variables are independent or dependent because in some circumstances part of the covariate may correlate. For the independent case, we set $\Sigma = I$. For the dependent case, we set $\Sigma_{ij} = 0.5^{i-j}$. To test the performance of our proposed algorithms under different situations, we set the dimension $p$ to be 20, 50 or 100, and the number of observations $n$ to be 100, 200 or 400. We consider three different tuning parameters $\lambda = 0.1, 1$ or 10.

3.1. Simulation for LASSO

We constructed a classical linear model where the vector of coefficients $\beta$ is $[1, -1, 2, -3, 1, 2, 0, \ldots, 0]^T$ (the first index represents the intercept in this linear model). We used the glmnet function embedded in R package glmnet to compare with our algorithm. This function use coordinate descent method which is stated by Friedman, Hastie and Tibshirani[12].

The $l_1$ norm of the difference between the results of two algorithms are shown in Table 1-6. The results show that ADMM can obtain ideal numerical convergence.

Table 1. The $l_1$ norm of the difference between the results of ADMM and glmnet ($\lambda = 0.1$, independent case)

|       | n=100    | n=200    | n=400    |
|-------|----------|----------|----------|
| $p=20$| 0.001413 | 0.000208 | $\leq 0.0001$ |
| $p=50$| 0.05676  | 0.004496 | 0.000751  |
| $p=100$| 1.438052 | 0.087495 | 0.010916  |

Table 2. The $l_1$ norm of the difference between the results of ADMM and glmnet ($\lambda = 1$, independent case)

|       | n=100    | n=200    | n=400    |
|-------|----------|----------|----------|
| $p=20$| 0.000389 | 0.000277 | $\leq 0.0001$ |
| $p=50$| 0.035364 | 0.009722 | 0.000729  |
| $p=100$| 0.214013 | 0.03883  | 0.009497  |

Table 3. The $l_1$ norm of the difference between the results of ADMM and glmnet ($\lambda = 10$, independent case)

|       | n=100    | n=200    | n=400    |
|-------|----------|----------|----------|
| $p=20$| $\leq 0.0001$ | 0.000107 | $\leq 0.0001$ |
| $p=50$| $\leq 0.0001$ | $\leq 0.0001$ | 0.000126  |
| $p=100$| 0.001315 | 0.000191 | 0.000323  |
Table 4. The $l_1$ norm of the difference between the results of ADMM and glmnet ($\lambda = 0.1$, dependent case)

| p   | n=100       | n=200       | n=400       |
|-----|-------------|-------------|-------------|
| 20  | 0.009231    | 0.004715    | 0.003296    |
| 50  | 0.07291     | 0.01915     | 0.012644    |
| 100 | 2.653374    | 0.135962    | 0.028054    |

Table 5. The $l_1$ norm of the difference between the results of ADMM and glmnet ($\lambda = 1$, dependent case)

| p   | n=100       | n=200       | n=400       |
|-----|-------------|-------------|-------------|
| 20  | 0.00498     | 0.003228    | 0.001594    |
| 50  | 0.014189    | 0.0118      | 0.004164    |
| 100 | 0.24706     | 0.055075    | 0.017698    |

Table 6. The $l_1$ norm of the difference between the results of ADMM and glmnet ($\lambda = 10$, dependent case)

| p   | n=100       | n=200       | n=400       |
|-----|-------------|-------------|-------------|
| 20  | 0.00087     | 0.00052     | 0.000958    |
| 50  | 0.00039     | 0.000518    | 0.001317    |
| 100 | 0.004306    | 0.001983    | 0.001537    |

The comparison tables of time cost are listed in Figure 1-6, where the black strips indicate the time cost of ADMM and the grey strips represent the time used by glmnet. The figures above imply that ADMM’s time cost is less than that of glmnet. These figures imply that ADMM’s time cost is less than that of glmnet in almost all the cases. Also, for fixed tuning parameter, as the number of coefficients and observations increase, the time cost of ADMM increase more slowly than glmnet.
3.2. Simulation for Group lasso

We use the same procedure as in LASSO to generate the dataset. However in this case we divide the covariates into some groups:

$$ \beta = (1,-1,2,-3,1,2,0,\ldots,0) = (1,\beta_1,\beta_2,\beta_3,\beta_4,\ldots) \quad (\beta_1 = (-1,2,-3), \beta_2 = (1,2), \beta_3 = (0,0,0), \beta_4 = (0,0), \beta_m = (0) \text{ for } m \geq 5) $$

$l_2$ penalty will be applied separately to those groups. Note that actually only the first two groups are nonzero vectors. The procedure of constructing the observations is the same as in LASSO, as stated above. The function `grplasso` in R package `grplasso` is applied. For the details of the algorithm, please refer to [13].
The \( l_1 \) norm of the difference between the results of two algorithms are shown below:

**Table 7.** The \( l_1 \) norm of the difference between the results of ADMM and \( \text{grplasso} \ (\lambda = 0.1, \ \text{independent case}) \)

|       | \( n=100 \)     | \( n=200 \)     | \( n=400 \)     |
|-------|-----------------|-----------------|-----------------|
| \( p=20 \) | 0.009978        | 0.004202        | 0.001662        |
| \( p=50 \) | 0.107016        | 0.016152        | 0.006910        |
| \( p=100 \) | 4.663680        | 0.078458        | 0.016133        |

**Table 8.** The \( l_1 \) norm of the difference between the results of ADMM and \( \text{grplasso} \ (\lambda = 1, \ \text{independent case}) \)

|       | \( n=100 \)     | \( n=200 \)     | \( n=400 \)     |
|-------|-----------------|-----------------|-----------------|
| \( p=20 \) | 0.074356        | 0.041666        | 0.014318        |
| \( p=50 \) | 0.585938        | 0.145715        | 0.066060        |
| \( p=100 \) | 2.255765        | 0.542820        | 0.156378        |

**Table 9.** The \( l_1 \) norm of the difference between the results of ADMM and \( \text{grplasso} \ (\lambda = 10, \ \text{independent case}) \)

|       | \( n=100 \)     | \( n=200 \)     | \( n=400 \)     |
|-------|-----------------|-----------------|-----------------|
| \( p=20 \) | 0.170725        | 0.222433        | 0.094887        |
| \( p=50 \) | 0.277922        | 0.359937        | 0.357325        |
| \( p=100 \) | 0.868632        | 0.766472        | 0.562727        |

**Table 10.** The \( l_1 \) norm of the difference between the results of ADMM and \( \text{grplasso} \ (\lambda = 0.1, \ \text{dependent case}) \)

|       | \( n=100 \)     | \( n=200 \)     | \( n=400 \)     |
|-------|-----------------|-----------------|-----------------|
| \( p=20 \) | 0.025212        | 0.006882        | 0.002378        |
| \( p=50 \) | 0.149601        | 0.025561        | 0.012626        |
| \( p=100 \) | 4.337529        | 0.139232        | 0.034336        |

**Table 11.** The \( l_1 \) norm of the difference between the results of ADMM and \( \text{grplasso} \ (\lambda = 1, \ \text{dependent case}) \)

|       | \( n=100 \)     | \( n=200 \)     | \( n=400 \)     |
|-------|-----------------|-----------------|-----------------|
| \( p=20 \) | 0.110155        | 0.064208        | 0.022173        |
| \( p=50 \) | 0.521091        | 0.210995        | 0.113988        |
| \( p=100 \) | 2.140328        | 0.696216        | 0.290701        |

**Table 12.** The \( l_1 \) norm of the difference between the results of ADMM and \( \text{grplasso} \ (\lambda = 10, \ \text{dependent case}) \)

|       | \( n=100 \)     | \( n=200 \)     | \( n=400 \)     |
|-------|-----------------|-----------------|-----------------|
| \( p=20 \) | 0.186464        | 0.214501        | 0.067447        |
| \( p=50 \) | 0.244576        | 0.379445        | 0.393245        |
| \( p=100 \) | 0.690592        | 0.742223        | 0.688781        |

The comparison tables of time cost are listed in Figure 7-12. Exact numerical values are in the appendix as in 3.1. In the table below the black bars still indicate the time cost of ADMM while the white bars still indicate the time cost of function \( \text{grplasso} \). As is shown by these figures, ADMM wins
against grplasso if we consider the time cost and the growing speed of time cost as the parameters increase.

Figure 7. The time cost of ADMM and grplasso (λ = 0.1, independent case)

Figure 8. The time cost of ADMM and grplasso (λ = 1, independent case)

Figure 9. The time cost of ADMM and grplasso (λ = 10, independent case)

Figure 10. The time cost of ADMM and grplasso (λ = 0.1, dependent case)
Figure 11. The time cost of ADMM and grplasso ($\lambda = 1$, dependent case)

Figure 12. The time cost of ADMM and grplasso ($\lambda = 10$, dependent case)

3.3. Simulation for Sparse inverse covariance estimation

Given a set of samples generated from multivariate Gaussian distribution $N(0, \Sigma)$, our goal is to estimate the precision matrix $\Sigma^{-1}$. In this example we let

$$
\Sigma_0 = \begin{cases} 
2.0 (i - j = 0) \\
0.6 (|i - j| = 1) \\
0.4 (|i - j| = 2) \\
0 (|i - j| \geq 3)
\end{cases}
$$

Note that in this section we do not need to consider the case that every component of the covariate is independent.

The function glasso in R package glasso is applied. This procedure is based on the paper of Friedman, Hastie and Tibshirani[3]. For the measurement of numerical accuracy we compute the difference matrix of two matrices obtained by ADMM and glasso and then check the elementwise $l_1$ norm of the difference matrix. From the tables below we could see ADMM achieves enough numerical accuracy.

Table 13. The $l_1$ norm of the difference between the results of ADMM and glasso ($\lambda = 0.1$)

|       | n=100  | n=200  | n=400  |
|-------|--------|--------|--------|
| $p=20$| 0.001509 | 0.001247 | 0.001146 |
| $p=50$| 0.009142 | 0.004697 | 0.003215 |
| $p=100$| 0.034976 | 0.014704 | 0.008182 |

Table 14. The $l_1$ norm of the difference between the results of ADMM and glasso ($\lambda = 1$)

|       | n=100  | n=200  | n=400  |
|-------|--------|--------|--------|
| $p=20$| 0.000049 | 0.000067 | 0.000055 |
| $p=50$| 0.000124 | 0.000132 | 0.000157 |
| $p=100$| 0.000194 | 0.000267 | 0.000290 |
Table 15. The $l_1$ norm of the difference between the results of ADMM and glasso ($\lambda = 10$)

|       | n=100   | n=200   | n=400   |
|-------|---------|---------|---------|
| $p=20$| 0.000018| 0.000020| 0.000016|
| $p=50$| 0.000049| 0.000035| 0.000045|
| $p=100$| 0.000066| 0.000078| 0.000094|

However, the simulation indicates that ADMM may cost more time than glasso to reach the ideal result. The time cost of matrix decomposition (see 2.28) may leads to low deficiency of ADMM.

4. Real-world data analysis

In this part we apply ADMM to deal with data from real world. The dataset we choose is birthwt in R package MASS. This dataset was collected during 1986 by Baystate Medical Center, Springfield[14]. The variables recorded in the dataset are presented in Table 16.

Table 16. Variables in birthwt dataset

| Variable name | Meaning           |
|---------------|-------------------|
| age           | mother's age in years. |
| lwt           | mother's weight in |
4.1 Linear regression analysis on birthwt with LASSO
We first applied lasso to find out the factors related to the birth weight of infants, and the ADMM method was used for the numerical solutions. We used 10-fold cross validation to determine the tuning parameter, and the R function cv.glmnet was implemented for this. For this dataset the tuning parameter was set as 39.3249.

We implement ADMM to solve the LASSO problem with tuning parameter $\lambda = 39.3249$, and the result are shown in Table 17. To check if the result is precise we also implement glmnet. As is indicated in table 17, the error is acceptable.

Table 17. Results for dataset birthwt via ADMM and glmnet, the results of glmnet are in the brackets.

| Variable | Results of ADMM | Results of glmnet | Numerical Error |
|----------|-----------------|-------------------|-----------------|
| intercept | 2375.853 | 2375.844 | 0.009 |
| age | 2.990352 | 2.990529 | 0.000177 |
| lwt | 3.794469 | 3.794497 | 0.000028 |
| smoke | -111.2471 | -111.2424 | 0.0047 |
| ptl | $\leq 10^{-8}$ | 0 | $\leq 10^{-8}$ |
| ht | $\leq 10^{-8}$ | 0 | $\leq 10^{-8}$ |
| ui | -136.2305 | -136.2310 | 0.0005 |
| ftv | $\leq 10^{-8}$ | 0 | $\leq 10^{-8}$ |
| race_white | 138.6107 | 138.6087 | 0.0020 |
| race_black | $\leq 10^{-8}$ | 0 | $\leq 10^{-8}$ |

Under LASSO the factors selected are age, lwt, smoke, ui, and race. So we can conclude that the age of mother, mother’s weight in last menstrual period, the presence of uterine irritability and the race of mother are the key factors.

4.2 Polynomial regression analysis on birthwt with group lasso
As all continuous functions can be approximated by polynomials, regression with polynomials may capture the nonlinear feature. We now analyze the dataset with polynomial regression. We use
polynomials of 3 degree to approximate the feature of mother’s age and mother’s weight at last menstrual time. To do this we set \( iage = age^i \) and \( ilwt = lwt^i \) \( (i = 1, 2, 3) \). And we select 3 variables of the form \( iage \) and 3 variables of the form \( ilwt \) separately as 2 groups. The discrete variable \( ptl \) and \( ftv \) are grouped separately with their dummy variables. In conclusion variables are grouped as below:

\[
\{age1, age2, age3\}, \{lwt1, lwt2, lwt3\}, \{white, black\}, \{smoke\}, \{ptl1, ptl2m\}, \{ht\}, \{ui\}, \{ftv1, ftv2, ftv3m\}.
\]

Then, we used group lasso to determine the key factors instead of simple LASSO. Similarly, a 10-fold cross validation is implemented and the tuning parameter is set as \( \lambda = 11.64447 \). The result is in the table 18.

| Variables | ADMM |
|-----------|------|
| intercept | 3.02225961 |
| age1      | 0.11423533 |
| age2      | 0.41561593 |
| age3      | 0.24677664 |
| lwt1      | 0.52619988 |
| lwt2      | -0.13975513 |
| lwt3      | 0.41530045 |
| white     | 0.15388875 |
| black     | -0.04713589 |
| smoke     | -0.16275857 |
| ptl1      | -0.14279620 |
| ptl2m     | 0.03469140 |
| ht        | -0.23939033 |
| ui        | -0.36043124 |
| ftv1      | 0 |
| ftv2      | 0 |
| ftv3m     | 0 |

Age was discovered to have nonlinear effects on the birth weight of the infant. Also, the number of physician visits during the first trimester is not a crucial factor, just as same as LASSO implies.

4.3. Detection of interaction between factors in birthwt with graphical lasso

In this section we try to detect the interaction between some factors that effect the infants’ weight using graphical lasso. The factors we select are: \( age, lwt, smoke, ptl, ht, ui, ftv \) and \( race \). As race takes only 3 values(1=white, 2=black, 3=others) we replace it with \( race_{white} \) and \( race_{black} \) as dummy variables. We apply ADMM to estimate the precision matrix with the tuning parameter set as \( \lambda = 0.1 \).

The obtained estimation of precision matrix is as follows:

|         | age | lwt | smoke | ptl | ht  | ui  | ftv | race_white | race_black |
|---------|-----|-----|-------|-----|-----|-----|-----|------------|------------|
| age     | 0.93| -0.07| 0     | 0   | 0   | 0   | -0.09| -0.09      | 0.01       |
The matrix indicates some underlying interaction between mother’s weight at last menstrual time and history of hypertension. Potential relationship is also found between 2 pairs: number of previous premature labors with presence of uterine irritability and smoking status during pregnancy with mother’s race. Note that the interaction between people of black race and white race is also implied in the matrix, but it is somewhat obvious because they are dummy variables related to the race of mother. Figure 16 provides a clearer view of the structure.

| lwt   | -0.07 | 0.94 | 0     | 0.03  | -0.11 | 0.04 | -0.03 | 0     | -0.11 |
|-------|-------|------|-------|-------|-------|------|-------|-------|-------|
| smoke | 0     | 0    | 0.95  | -0.07 | 0     | 0    | 0     | -0.18 | 0     |
| pti   | 0     | 0.03 | -0.07 | 0.93  | 0     | -0.11| 0     | 0     | 0     |
| ht    | 0     | -0.11| 0     | 0     | 0.92  | 0    | 0     | 0     | 0     |
| ui    | 0     | 0.04 | 0     | -0.11| 0     | 0.92 | 0     | 0     | 0     |
| ftv   | -0.09 | -0.03 | 0   | 0     | 0     | 0    | 0.92  | 0    | 0     |
| race_white | -0.09 | 0     | -0.18| 0     | 0     | 0    | 0     | 1.03  | 0.27  |
| race_b_lack | 0.01 | -0.11| 0     | 0     | 0     | 0    | 0     | 0.27  | 1     |
Figure 16: Undirected graph from graphical lasso. The nodes are:
1: age, 2: lwt, 3: smoke, 4: ptl, 5: ht, 6: ui, 7: ftv, 8: race_white, 9: race_black

5. Conclusion
In this paper we state that ADMM is an appropriate algorithm for $l_1$ penalized problem. By simulation we show that ADMM is competitive among other commonly used algorithm we regard to numerical accuracy and time cost. Although ADMM may not be the best choice in every situation (as is shown in 3.3), there are many situations such that ADMM performs ideally. Finally we apply ADMM to analyze the real-world dataset birthwt, which is about birth weight of infant. LASSO, group lasso and graphical lasso, associated with ADMM to solve themselves, are implemented to capture the linear and nonlinear feature inside the model and the potential interaction between some factors.

References
[1] Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. *Journal of The Royal Statistical Society Series B-statistical Methodology*, 73(3), 273-282.
[2] Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of The Royal Statistical Society Series B-statistical Methodology*, 68(1), 49-67.
[3] Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of The Royal Statistical Society Series B-statistical Methodology*, 68(1), 49-67.
[4] Efron, B., Hastie, T., Johnstone, I. M., Tibshirani, R., Ishwaran, H., Knight, K., ... & Weisberg, S. Least angle regression. *Annals of Statistics*, 32(2), 407-499. EPL, 84 (2008)
[5] Fu, W. J. (1998). Penalized Regressions: The Bridge versus the Lasso. *Journal of Computational and Graphical Statistics*, 7(3), 397-416.
[6] Lauritzen, S. (2004). *Graphical models*. Oxford: Clarendon Press.
[7] Banerjee, O., Ghaoui, L. E., & Daspremont, A. (2008). Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data. *Journal of Machine Learning Research.*, 485-516.
[8] Yuan, M., & Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1), 19-35.
[9] Gabay, D., & Mercier, B. (1976). A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1), 17-40.
[10] Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations & Trends in Machine Learning, 3*(1), 1-122.

[11] Bo Geng, Yangxi Li, Dacheng Tao, Meng Wang, Zheng-Jun Zha, & Chao Xu. (2012). Parallel lasso for large-scale video concept detection. *IEEE Transactions on Multimedia, 14*(1), 55-65.

[12] Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software, 33*(01), 1-22.

[13] Lukas Meier, Sara van de Geer, & Peter Bühlmann. (2010). The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B Statistical Methodology, 70*(1), 53-71.

[14] Hosmer, D. W., & Lemeshow, S. (2000). *Applied logistic regression*. J. Wiley.