MATH 680 Computation Intensive Statistics

Final Project

Estimation and Variable Selection in Linear and Logistic Regression Using Alternating Direction Method of Multipliers (ADMM)

Shouao Wang
Yi Lian

Department of Epidemiology, Biostatistics and Occupational Health
Faculty of Medicine
McGill University

Supervised by
Dr. Yi Yang

Department of Mathematics and Statistics
Faculty of Science
McGill University

Fall 2016

INTRODUCTION

Alternating direction method of multipliers (ADMM) is a relatively new class of algorithms that aims to solve convex optimization problems in statistics and machine learning (Boyd et al., 2011). It is based on two previous algorithms - dual (sub)gradient descent method and augmented Lagrangian method. In a nutshell, ADMM takes advantage of the better decomposability of dual (sub)gradient descent method and the superior convergence property of the augmented Lagrangian method (Boyd et al., 2011). This will be explained in more details in the **METHODS** section with respect to group LASSO penalty.

Logistic regression is a classic regression model (Cox, 1958). It is a special case of generalized linear model where the dependent variable is categorical. This special characteristic makes logistic regression very popular in biostatistics and epidemiology, where investigators are working mostly with binary outcomes, e.g., death vs. survival and HIV+ vs. HIV-. In spite of the unsettled dispute over the odds ratio (OR) in biostatistics and epidemiology, logistic regression has been proven to be a robust and efficient statistical tool for biostatistical inference and risk prediction.

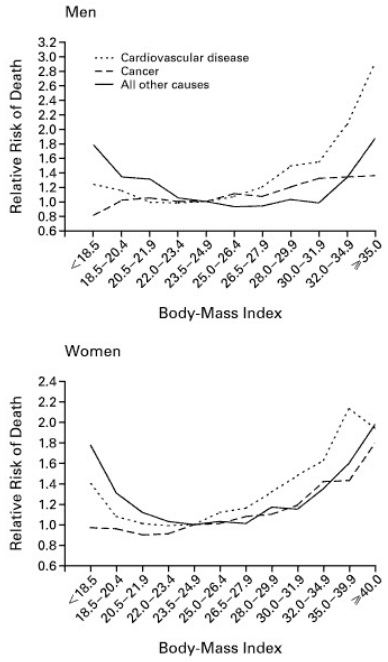

Figure 1: Multivariate Relative Risk of Death from Cardiovascular Disease, Cancer, and All Other Causes among Men and Women Who Had Never Smoked and Who Had No History of Disease at Enrollment, According to Body-Mass Index (Calle et al., 1999).

Beside the outcomes, exposure definition also makes biostatistics somehow unique comparing to other applications of statistics. A main reason is that the effects of some common independent variables (patient characteristics, confounding variables, etc.) such as age and body mass index (BMI) are not linear. As shown in

Figure 1 (Calle et al., 1999), the analysis revealed a U-shaped relationship between BMI and the risk of death from cardiovascular disease, cancer and all other causes. This pattern is biologically plausible as people with BMI around 20-25 are usually considered "healthy" and are therefore at the lowest risk of cardiovascular diseases.

Biostatisticians have adopted different methods, e.g. splines, to model such non-linear associations. However, categorization (or stratification) is always one of the most simple and intuitive ways to go. Let us go back to the example in Figure 1, where BMI is categorized into 12 groups. Comparing to other methods, categorization provides a cleaner estimation of association with virtually no assumptions, at the cost of only a few degrees of freedom. In terms of programming, instead of a linear term for BMI, we have $12 - 1 = 11$ independent dummy variables. For each of the categories, we are able to calculate the odds ratio with 95% confidence intervals. Things are working well for inference.

However, if we want to conduct risk prediction based on the fitted logistic regression model and would like to exclude a few variables for simplicity, problems will rise. As the dummy variables are independent, some of them may be excluded by some common variable selection methods such as stepwise selection and LASSO. The exclusion of any of the dummy variables will lead to failure in predicting the outcome probability for individuals in the corresponding category. Therefore, additional methods that select groups of variables instead of individual variables are needed. Group LASSO method was developed to solve the problem (Bakin et al., 1999; Yuan and Lin, 2006).

Computationally, the group LASSO is more challenging than the LASSO (Yang and Zou, 2015). A number of different algorithms have been proposed to solve the LASSO penalized least squares (Yang and Zou, 2015) including least angle regression (LARS) algorithm (Efron et al., 2004) and coordinate descent algorithm (Tseng, 2001; Fu, 1998). As Yang and Zou (2015) pointed out that LARS-type algorithms cannot be applied to group LASSO penalty. Later, a block-wise descent algorithm for group LASSO was proposed by Yuan and Lin (2006) following Fu (1998). However, this algorithm requires the group-wise orthonormal condition that is not desirable. In this report, we will derive an algorithm to solve the group-LASSO penalized loss functions (square loss and logistic loss) using ADMM.

METHODS

Overview

We developed an `R` package `LassoADMM` to perform estimation and variable selection in linear and logistic regression with LASSO and group LASSO penalty. Optimization problems with LASSO penalty can be solved with various well-established algorithms including LARS and (sub)gradient descent. Therefore, the focus of this project is the optimization of square and logistic loss with group LASSO penalty.

As mentioned in the **INTRODUCTION** section, ADMM is an algorithm that is intended to blend the decomposability of dual (sub)gradient descent with the superior convergence properties of the method of multipliers (Boyd et al., 2011). In the case of group LASSO penalty, the objective function is separable therefore can be solved in parallel with the dual (sub)gradient descent methods (dual decomposition). This property is very much desired. However, these methods come with an obvious disadvantage, which is poor convergence properties. In ADMM, convergence property is improved by utilizing augmented Lagrangian.

The algorithm solves problems in the form

$$\text{minimize } f(x) + g(x), \text{ subject to } Ax + Bz = c$$

The augmented Lagrangian is

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + (\rho/2)\|Ax + Bz - c\|_2^2$$

With the scaled form, the ADMM is:

$$x^{k+1} = \underset{x}{\operatorname{argmin}}(f(x) + (\rho/2)\|Ax^k + Bz^k + u^k - c)\|_2^2$$

$$x^{k+1} = \underset{z}{\operatorname{argmin}}(g(z) + (\rho/2)\|Ax^{k+1} + Bz^k + u^k - c)\|_2^2$$

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

Define dual residual at iteration $k+1$ as $s^{k+1} = \rho A^T B(z^{k+1} - z^k)$.

And primal residual at iteration $k+1$ as $r^{k+1} = Ax^{k+1} + Bz^{k+1} - c$.

Stop the algorithm when the dual and primal residuals are small enough.

Algorithms

For simplicity, we will only consider the case with no intercept. The case with intercept will be similar and easy to implement.

1. LASSO - Square Loss

Consider the problem:

$$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

Rewrite above as $\min_{\beta,\alpha} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\alpha\|_1$, subject to $\beta - \alpha = 0$.

The Algorithm is:

Step 1: Initialize β, α, ω

Step 2: Update β, α, ω

$$\begin{aligned}
\beta^{(k+1)} &= \arg\min_{\beta} \frac{1}{2}\|y - X\beta^{(k)}\|_2^2 + \frac{\rho}{2}\|\beta^{(k)} - \alpha^{(k)} + \omega^{(k)}\|_2^2 \\
&= (X^T X + \rho I)^{-1}(X^T y + \rho(\alpha^{(k)} - \omega^{(k)})) \\
\alpha^{(k+1)} &= \arg\min_{\alpha} \lambda\|\alpha^{(k)}\|_1 + \frac{\rho}{2}\|\beta^{(k+1)} - \alpha^{(k)} + \omega^{(k)}\|_2^2 \\
&= S_{\lambda/\rho}(\beta^{(k+1)} + \omega^{(k)}) \\
\omega^{(k+1)} &= \omega^{(k)} + \beta^{(k+1)} - \alpha^{(k+1)}
\end{aligned}$$

Step 3: Stop when converge.

Dual residual: $s^{k+1} = \rho A^T B(z^{k+1} - z^z) = \rho(\alpha^{(k+1)} - \alpha^{(k)})$

Primal residual $r^{k+1} = Ax^{k+1} + Bz^{k+1} - C = \beta^{(k+1)} - \alpha^{(k+1)}$.

Stop when both dual and primal residuals are small enough.

Note $S_t(a)$ is the soft thresholding operator.

$$S_t(x) = \text{sign}(x)(|x| - t)_+$$

The β-update is essentially a ridge regression (i.e. quadratically regularized least squares) computation, so ADMM can be interpreted as a method for solving the lasso problem by iteratively carrying out ridge regression.

Also, the choice of ρ can greatly effect practical convergence of ADMM, if ρ is too large, then not enough emphasis on minimizing $f+g$. If ρ is too small, then not enough emphasis on feasibility. More details of strategy for choose ρ can be find in Boyd et al. (2010) "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers".

2. LASSO - Logistic Loss

Consider the problem:

$$\min_{\beta} \sum \log[1+\exp(-y \cdot X\beta)] + \lambda\|\beta\|_1$$

Rewrite above as $\min_{\beta,\alpha} \sum \log[1+\exp(-y \cdot X\beta)] + \lambda\|\beta\|_1$, subject to $\beta - \alpha = 0$

The Algorithm is:

Step 1: Initialize β, α, ω

Step 2: Update β, α, ω

$$\beta^{(k+1)} = \arg\min_{\beta} \sum \log[1+\exp(-y \cdot X\beta^{(k)})] + \frac{\rho}{2}\|\beta^{(k)} - \alpha^{(k)} + \omega^{(k)}\|_2^2$$

$$\alpha^{(k+1)} = S_{\lambda/\rho}(\beta^{(k+1)} + \omega^{(k)})$$

$$\omega^{(k+1)} = \omega^{(k)} + \beta^{(k+1)} - \alpha^{(k+1)}$$

Step 3: Stop when converge.

The β update could be solved using L-BFGS or Newton's method. Here we will use Newton's method, let

$$f = \sum \log[1+\exp(-y \cdot X\beta^{(k)})] + \frac{\rho}{2}\|\beta^{(k)} - \alpha^{(k)} + \omega^{(k)}\|_2^2$$

The gradient of f is:

$$\nabla f = \sum(-yX) \cdot \frac{\exp(-yX\beta)}{1+\exp(-yX\beta)} + \rho(\beta - \alpha + \omega)$$

The Hessian of f is:

$$\nabla^2 f = (-yX) \cdot \frac{\exp(-yX\beta)}{(1+\exp(-yX\beta))^2} \cdot (-yX) + \rho I$$

Hence we could update β as $\beta^{(k+1)} = \beta^{(k)} - \nabla f(\beta^{(k)})[\nabla^2 f(\beta^{(k)})]^{-1}$

3. Group LASSO - Square Loss

Consider the problem:

$$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda \sum_{g=1}^{G} c_g \|\beta\|_2$$

Rewrite above as $\min\limits_{\beta,\alpha} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda \sum_{g=1}^{G} c_g \|\alpha\|_2$, subject to $\beta - \alpha = 0$.

The Algorithm is :

Step 1: Initialize β, α, ω

Step 2: Update β, α, ω

$$\beta^{(k+1)} = (X^T X + \rho I)^{-1}(X^T y + \rho(\alpha^{(k)} - \omega^{(k)}))$$
$$\alpha_{(g)}^{(k+1)} = R_{c_g \lambda/\rho}(\beta_{(g)}^{(k+1)} + \omega_{(g)}^{(k)})$$
$$\omega^{(k+1)} = \omega^{(k)} + \beta^{(k+1)} - \alpha^{(k+1)}$$
$$g = 1, ..., G$$

Step 3: Stop when converge.

Notice the above algorithm is almost the same as lasso regression, the only difference is the α

update. And R_t^x is the group soft thresholding operator defined as:

$$R_t(x) = (1 - \frac{t}{\|x\|_2})_+ x$$

This group soft thresholding operator reduces to the soft thresholding operator when x is a scalar.

4. Group LASSO - Logistic Loss

Consider the problem:

$$\min_{\beta} \sum \log[1 + \exp(-y \cdot X\beta)] + \lambda \sum_{g=1}^{G} c_g \|\beta_{(g)}\|_2$$

Rewrite above as $\min_{\beta,\alpha} \sum \log[1 + \exp(-y \cdot X\beta)] + \lambda \sum_{g=1}^{G} c_g \|\alpha_{(g)}\|_2$, subject to $\beta - \alpha = 0$.

The algorithm is:

Step 1: Initialize β, α, ω

Step 2: Update β, α, ω:

$$\beta^{(k+1)} = \arg\min_{\beta} \sum \log[1 + \exp(-y \cdot X\beta^{(k)})] + \frac{\rho}{2} \|\beta^{(k)} - \alpha^{(k)} + \omega^{(k)}\|_2^2$$

$$\alpha_{(g)}^{(k+1)} = R_{c_g \lambda/\rho}(\beta_{(g)}^{(k+1)} + \omega_{(g)}^{(k)})$$

$$\omega^{(k+1)} = \omega^{(k)} + \beta^{(k+1)} - \alpha^{(k+1)}$$

$$g = 1, ..., G$$

Step 3: Stop when converge.

View the above algorithm as a combination of group penalize and logistic loss function, which will be similar as above.

SIMULATION RESULTS

The simulation is based on the following set up:

- $n = 100$, $p = 20$, $X \sim N_{n \cdot p}(0, 1)$.
- $\beta_1 = \beta_2 = ... = \beta_{20} = 1$, assuming there is no intercept.
- For the continous response (the square loss function case), $y = X\beta + \epsilon$, where $\epsilon \sim N_n(0, 1)$.
- For the binary response (the logistic loss function cass), $z = X\beta + \epsilon$, $p = \exp(z)/(1 + \exp(z))$, and $y \sim Bernoulli(p)$.
- For the group, we assign the first 5 covariates as group 1, the second 5 covariates as group 2, the third 5 covariates as group 3, the last 5 covariates as group 4.

1. LASSO - Square Loss

```
library(package="LassoADMM")
set.seed(680)
n<-100; p<-20
X <- matrix(rnorm(n*p), nrow=n, ncol=p)
X[,1]<-1
beta.star<-rep(1,p)
sig<-1
y<- X%*%beta.star+sig*rnorm(n, 0, 1)
admmlasso_ls(X=X,y=y,lam=0.01)

## $beta0
##           [,1]
## [1,] 0.8804912
##
## $beta
##  [1] 1.0407809 1.0331822 1.1180896 0.9872600 0.9103182 1.1274372 0.9606507
##  [8] 1.1109741 1.0461683 1.0582672 1.0737253 0.7734397 1.0997387 1.1352443
## [15] 1.1542025 0.9684015 1.0434475 0.8617243 0.9780819
##
## $total.iterations
## [1] 10
```

2. LASSO - Logistic Loss

```
set.seed(680)
X = matrix(rnorm(n*p),nrow=n,ncol=p)
beta.star=rep(1,p)
z = X%*%beta.star         # linear combination with a bias
pr = 1/(1+exp(-z))          # pass through an inv-logit function
y = rbinom(n,1,pr)
admmlasso_log(X=X, y=y, lam=0.01)

## $beta
##  [1]  0.9650758  0.9611866  0.9601567  0.9604790 -0.9585979  0.9596552
##  [7]  0.9623677  0.9635159  0.9596049  0.9603513  0.9593895  0.9615325
## [13]  0.9584545  0.9619146 -0.9582395  0.9607517  0.9614599  0.9592971
## [19]  0.9583942  0.9592976
##
## $total.iterations
## [1] 114
```

3. Group LASSO - Square Loss

```
set.seed(680)
n<-100; p<-20
X <- matrix(rnorm(n*p), nrow=n, ncol=p)
beta.star<-rep(1,p)
sig<-1
y<- X%*%beta.star+sig*rnorm(n, 0, 1)
group <- rep(1:4,each=5)
admmgrouplasso_ls(X,y,group=group,lam=0.01)

## $beta
##  [1] 1.0182212 1.0425464 1.0358311 1.1167698 0.9902237 0.9109811 1.1283215
##  [8] 0.9585878 1.1098355 1.0431568 1.0558078 1.0734974 0.7714324 1.0997284
## [15] 1.1350320 1.1489122 0.9668785 1.0432639 0.8596143 0.9774510
##
## $total.iterations
## [1] 10
```

4. Group LASSO - Logistic Loss

```
set.seed(680)
n<-100; p<-20
X <- matrix(rnorm(n*p), nrow=n, ncol=p)
```

```r
beta.star<-rep(1,p)
z<- X%*%beta.star
pr = exp(z)/(1+exp(z))          # pass through an inv-logit function
y = rbinom(n,1,pr)
group <- rep(1:4,each=5)
admmgrouplasso_log(X,y,group=group,lam=0.01)

## $beta
##  [1]  4.497157211  2.092145211  1.455293446  1.654577159 -0.491335839
##  [6] -0.007956819 -0.019611342 -0.024544559 -0.007740788 -0.010947598
## [11] -0.006815369 -0.016022511 -0.002797918 -0.017664617  0.001874080
## [16] -0.012668048 -0.015710835 -0.006418329 -0.002538619 -0.006420167
##
## $total.iterations
## [1] 230
```

From the simulation, the estimates are close to the pre-setup values of parameter. And the group lasso with logistic loss selects the first group.

DISCUSSION

The aim of this project is to learn ADMM through this lasso example with different setups, yet there is still something we need further thinking; for example,

- The cross validation for appropriate ρ and λ.
- How to guarantee the sparsity property of LASSO, since the update of β does not quite explain that.
- We could extend the study of logistic LASSO via ADMM to the case of using logistic LASSO as a classifier.

Bibliography

Bakin, S. et al. (1999). Adaptive regression and model selection in data mining problems.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122.

Calle, E. E., Thun, M. J., Petrelli, J. M., Rodriguez, C., and Heath Jr, C. W. (1999). Body-mass index and mortality in a prospective cohort of us adults. *New England Journal of Medicine*, 341(15):1097–1105.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 215–242.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). Least angle regression. *The Annals of statistics*, 32(2):407–499.

Fu, W. J. (1998). Penalized regressions: the bridge versus the lasso. *Journal of computational and graphical statistics*, 7(3):397–416.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494.

Yang, Y. and Zou, H. (2015). A fast unified algorithm for solving group-lasso penalize learning problems. *Statistics and Computing*, 25(6):1129–1141.

Yuan, M. and 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.