# Regularized Precision Matrix Estimation via ADMM

*Matt Galloway*

*February 23, 2018*

## Contents



This note describes the estimation procedure for regularized precision matrices using the alternating direction method of multipliers (ADMM) [1]. We use an elastic-net-type penalty that allows for flexible experimentation.

(WE MIGHT WANT TO EXPLAIN DUAL ASCENT FURTHER)

# 1 Introduction

Suppose we want to solve the following optimization problem. . .

Solve blah blah blah. . . . subject to

Suppose we want to minimize $f(x) + g(z)$ subject to the constraint that $Ax + Bz = c$. For now, we will take $x \in \mathbb{R}^n, z \in \mathbb{R}^m, A \in \mathbb{R}^{p \times m}, B \in \mathbb{R}^{p \times m}, c \in \mathbb{R}^p$ – though we will later consider cases where $x$ and $z$ are matrices. The *augmented lagrangian* is constructed as follows:

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}\left\|Ax + Bz - c\right\|_2^2$$

where $y \in \mathbb{R}^p$ is the lagrange multiplier. The optimal value is

$$p^* = inf\left\{f(x) + g(z)|Ax + Bz = c\right\}$$

Clearly, the minimization problem under the augmented lagrangian (RE-WORK) is equivalent to that of the usual lagrangian since any feasible point $(x, z)$ satisfies the constraint $\rho\left\|Ax + Bz - c\right\|_2^2/2 = 0$.

The ADMM algorithm consists of the following repeated iterations:

$$x^{k+1} := \arg\min_x L_\rho(x, z^k, y^k) \tag{1}$$

$$z^{k+1} := \arg\min_z L_\rho(z^{k+1}, z, y^k) \tag{2}$$

$$y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \tag{3}$$

A more complete introduction to the algorithm – specifically how it arose out of *dual ascent* and *method of multipliers* – can be found in Boyd, et al. (2011).

## 2 Regularized Precision Matrix Estimation

We now consider the case where $X_1, ..., X_n$ are iid $N_p(\mu, \Sigma)$ and we are tasked with estimating the precision matrix, denoted $\Omega \equiv \Sigma^{-1}$. The maximum likelihood estimator for $\Omega$ is

$$\hat{\Omega} = \arg\min_{\Omega \in S_+^p} \left\{ Tr\left(S\Omega\right) - \log\det\left(\Omega\right) \right\}$$

where $S = \sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})^T/n$. It is straight forward to show that when the solution exists, $\hat{\Omega} = S^{-1}$.

We can further construct a penalized likelihood estimator by adding a penalty term, $P_\lambda\left(\Omega\right)$, to the likelihood:

$$\hat{\Omega}_\lambda = \arg\min_{\Omega \in S_+^p} \left\{ Tr\left(S\Omega\right) - \log\det\left(\Omega\right) + P_\lambda\left(\Omega\right) \right\}$$

Throughout the rest of this document we will take $P_\lambda\left(\Omega\right)$ to be $P_\lambda\left(\Omega\right) = \lambda\left[\frac{1-\alpha}{2}\left\|\Omega\right|_F^2 + \alpha\left\|\Omega\right\|_1\right]$ so that the full penalized likelihood is as follows:

$$\hat{\Omega}_\lambda = \arg\min_{\Omega \in S_+^p} \left\{ Tr\left(S\Omega\right) - \log\det\left(\Omega\right) + \lambda\left[\frac{1-\alpha}{2}\left\|\Omega\right|_F^2 + \alpha\left\|\Omega\right\|_1\right] \right\}$$

where $0 \leq \alpha \leq 1$, $\lambda > 0$, $0 < \eta < 2$, $\left\|\cdot\right\|_F^2$ is the Frobenius norm and we define $\left\|A\right\|_1 = \sum_{i,j}\left|A_{ij}\right|$. This penalty is closely related to the elastic-net penalty explored by Hui Zou and Trevor Hastie [4]. Clearly, when $\alpha = 0$ this reduces to a ridge-type penalty and when $\alpha = 1$ this reduces to a lasso-type penalty.

By letting $f$ be equal to the non-penalized likelihood and $g$ equal to $P_\lambda\left(\Omega\right)$, our goal is to minimize the full augmented lagrangian where the constraint is that $\Omega - Z$ is equal to zero:

$$L_\rho(\Omega, Z, \Lambda) = f\left(\Omega\right) + g\left(Z\right) + Tr\left[\Lambda\left(\Omega - Z\right)\right] + \frac{\rho}{2}\left\|\Omega - Z\right\|_F^2$$

The ADMM algorithm for regularized precision matrix estimation is

$$\Omega^{k+1} = \arg\min_{S\Omega}\left\{ Tr\left(\Omega\right) - \log\det\left(\Omega\right) + Tr\left[\Lambda^k\left(\Omega - Z^k\right)\right] + \frac{\rho}{2}\left\|\Omega - Z^k\right\|_F^2 \right\} \tag{4}$$

$$Z^{k+1} = \arg\min_{Z}\left\{ \lambda\left[\frac{1-\alpha}{2}\left\|Z\right\|_F^2 + \alpha\left\|Z\right\|_1\right] + Tr\left[\Lambda^k\left(\Omega^{k+1} - Z\right)\right] + \frac{\rho}{2}\left\|\Omega^{k+1} - Z\right\|_F^2 \right\} \tag{5}$$

$$\Lambda^{k+1} = \Lambda^k + \rho\left(\Omega^{k+1} - Z^{k+1}\right) \tag{6}$$

We can simplify this algorithm further using the *condensed-form ADMM* which we will show in the next section.

## 2.1 Condensed-Form ADMM

A sometimes more convenient form of the ADMM algorithm is constructed by scaling the dual variable. Let us define $R^k = \Omega - Z^k$ and $U^k = \Lambda^k/\rho$. Then

$$
\begin{aligned}
Tr\left[\Lambda^k\left(\Omega - Z^k\right)\right] + \frac{\rho}{2}\left\|\Omega - Z^k\right\|_F^2 &= Tr\left[\Lambda^k R^k\right] + \frac{\rho}{2}\left\|R^k\right\|_F^2 \\
&= \frac{\rho}{2}\left\|R^k + \Lambda^k/\rho\right\|_F^2 - \frac{\rho}{2}\left\|\Lambda^k/\rho\right\|_F^2 \\
&= \frac{\rho}{2}\left\|R^k + U^k\right\|_F^2 - \frac{\rho}{2}\left\|U^k\right\|_F^2
\end{aligned}
$$

**The condensed-form can now be written as follows:**

$$
\Omega^{k+1} = \arg\min_{\Omega}\left\{Tr\left(\Omega\right) - \log\det\left(\Omega\right) + \frac{\rho}{2}\left\|\Omega - Z^k + U^k\right\|_F^2\right\} \tag{7}
$$

$$
Z^{k+1} = \arg\min_{Z}\left\{\lambda\left[\frac{1-\alpha}{2}\left\|Z\right\|_F^2 + \alpha\left\|Z\right\|_1\right] + \frac{\rho}{2}\left\|\Omega^{k+1} - Z + U^k\right\|_F^2\right\} \tag{8}
$$

$$
U^{k+1} = U^k + \Omega^{k+1} - Z^{k+1} \tag{9}
$$

More generally (in vector form),

$$
x^{k+1} := \arg\min_{x}\left\{f(x) + \frac{\rho}{2}\left\|Ax + Bz^k - c + u^k\right\|_2^2\right\} \tag{10}
$$

$$
z^{k+1} := \arg\min_{z}\left\{g(z) + \frac{\rho}{2}\left\|Ax^{k+1} + Bz - c + u^k\right\|_2^2\right\} \tag{11}
$$

$$
u^{k+1} := u^k + Ax^{k+1} + Bz^{k+1} - c \tag{12}
$$

Note that there are limitations to using this method. For instance, because the dual variable is scaled by $\rho$ (the step size), this form limits one to using a constant step size (without making further adjustments to $U^k$) – a limitation that could prolong the convergence rate.

## 2.2 Algorithm

$$
\begin{aligned}
\Omega^{k+1} &= \arg\min_{\Omega} \left\{ Tr\left(\Omega\right) - \log\det\left(\Omega\right) + \frac{\rho}{2}\left\|\Omega - Z^{k} + U^{k}\right\|_{F}^{2} \right\} \\
Z^{k+1} &= \arg\min_{Z} \left\{ \lambda \left[ \frac{1-\alpha}{2}\left\|Z\right\|_{F}^{2} + \alpha\left\|Z\right\|_{1} \right] + \frac{\rho}{2}\left\|\Omega^{k+1} - Z + U^{k}\right\|_{F}^{2} \right\} \\
U^{k+1} &= U^{k} + \Omega^{k+1} - Z^{k+1}
\end{aligned}
$$

1. Decompose $S + \rho(U^{k} - Z^{k}) = VQV^{T}$.

$$
\Omega^{k+1} = \frac{1}{2\rho} V \left[ -Q + \left( Q^{2} + 4\rho I_{p} \right)^{1/2} \right] V^{T}
$$

2. Elementwise soft-thresholding for all $i = 1, ..., p$ and $j = 1, ..., p$.

$$
\begin{aligned}
Z_{ij}^{k+1} &= \frac{1}{\lambda(1-\alpha)+\rho} sign\left( \Omega_{ij}^{k+1} + U_{ij}^{k} \right) \left( \rho\left| \Omega_{ij}^{k+1} + U_{ij}^{k} \right| - \lambda\eta\alpha \right)_{+} \\
&= \frac{1}{\lambda(1-\alpha)+\rho} Soft\left( \rho(\Omega_{ij}^{k+1} + U_{ij}^{k}), \lambda\eta\alpha \right)
\end{aligned}
$$

3. Update $U$.

$$
U^{k+1} = U^{k} + \Omega^{k+1} - Z^{k+1}
$$

### 2.2.1 Proof:

(Work in progress.)

## References


[1] Boyd, Stephen, et al. "Distributed optimization and statistical learning via the alternating direction method of multipliers." Foundations and Trends® in Machine Learning 3.1 (2011): 1-122.

[2] Polson, Nicholas G., James G. Scott, and Brandon T. Willard. "Proximal algorithms in statistics and machine learning." Statistical Science 30.4 (2015): 559-581.

[3] Marjanovic, Goran, and Victor Solo. "On $l_q$ optimization and matrix completion." IEEE Transactions on signal processing 60.11 (2012): 5714-5724.

[4] Zou, Hui, and Trevor Hastie. "Regularization and variable selection via the elastic net." Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67.2 (2005): 301-320.

# 3 Appendix