Inverse Covariance Selection via ADMM

Arturo Arranz Mateo

November 30, 2017

Abstract

In a Gaussian distributions, conditional independence between variables correspond to zero entries in the inverse covariance matrix. However, estimating the inverse from samples under the assumption of sparsity is not straight forward. For small matrices an fixed sparse pattern, it can be calculated as a convex optimization problem. Nonetheless, for high dimensional data and general sparse patterns the problem becomes intractable and heuristics are needed, such as lasso regularization. Another associated problem of with high dimensional data, which accounts for most of the data applications nowadays, is the problem of storage and computation time. In this project explore the distributed algorithm, Alternating Directions Method of Multipliers(ADMM) for the *inverse covariance selection problem* thanks to its decomposability nature and finally we show some applications for political voting analysis in the US senate.

1 Alternating Direction Method of Multipliers

The main foundations of ADMM are *the dual ascent method* for solving convex optimization problems, *dual decomposition* for decomposing the objective and constraint functions and *the method of multipliers* which guarantee differentiability under mild conditions.

1.1 Dual Ascent

Given the following convex optimization problem with linear constraints,

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array} \tag{1.1}$$

with variables $x \in \mathbb{R}^n$, where $A \in \mathbb{R}^{m \times n}$, and $f : \mathbb{R}^n \to \mathbb{R}$ is convex. The lagrangian,

$$L(x, y) = f(x) + y^T(Ax - b)$$

and the dual function,

$$g(y) = \inf_x L(x, y) = -f * (-A^T y) - b^T y$$

where $y \in \mathbb{R}^m$ is the Lagrangian multiplier, and f^* is the convex conjugate of f. The dual problem then becomes

$$\text{minimize} \quad g(y)$$

if strong duality holds, then we can recover the solution of the primal problem,x^* from the the dual problem solution, y^* as

$$x^* := minimize L(x, y^*)$$

the *dual ascent method* consists on taking steps towards the dual function gradient and updating the primal solution iteratively,

$$x^{k+1} := \operatorname*{argmin}_x L(x, y^k) \tag{1.2}$$

$$y^{k+1} := y^k + \alpha^k(Ax^{k+1} - b) \tag{1.3}$$

where $\alpha^k > 0$ is the step size, which control how fast the solution converges, but in case of too high value it might lead to convergence problems. The names *dual ascent* comes from the fact that $g(y)$ increases in every step. However, some conditions must hold to guarantee convergence and functioning of the algorithm. For instance, $g(y)$ must be differentiable in order to evaluate the it gradient, and *dual subgradient method* should be used instead. Another exampel is the case when $f(x)$ is a non-zero affine function, in which case the update (1.2) would fail since the Lagrangian is unbounded below for most of y and x.

1.2 Dual decomposition

The strength of *dual ascent* is that it can lead to a decentralized algorithm when the objective and constraint functions are separable,

$$f(x) = \sum_{i=1}^{N} f_i(x_i) \qquad Ax = \sum_{i=1}^{N} A_i x_i$$

where $x = (x_1, x_2, ...x_N)$ and $x_i \in \mathbb{R}^{n_i}$. Then the Lagrangian looks like

$$L(x, y) = \sum_{i=1}^{N} f_i(x_i) + y^T(A_i x_i - b) - (1/N)y^T b$$

meaning that the dual ascent method consist in N primal variables updates and one dual step as

$$x_i^{k+1} := \underset{x}{\text{argmin}} \quad L_i(x_i, y^k) \tag{1.4}$$

$$y^{k+1} := y^k + \alpha^k(Ax^{k+1} - b) \tag{1.5}$$

1.3 Method of multipliers

As already stated, dual ascent assume very strict conditions on the initial problem such as strict convexity and finiteness of f. In order to overcome some of this issues the *augmented Lagrangian* is introduced,

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

where ρ is the *penalty parameter*. The *augmented Lagrangian* is equivalent to solve the transformed initial problem

$$\begin{array}{ll} \text{minimize} & f(x) + ||Ax - b||_2^2 \\ \text{subject to} & Ax = b \end{array} \tag{1.6}$$

The problem (1.6) is clearly equivalent to the original problem, (1.1), since the residual $Ax - b$ is zero at the optimal feasible point. This new formulation brings the benefit of differentiability under rather mild conditions on the original problem. The new dual functions is $g_\rho(y) = inf L_\rho(x, y)$. This leads to the *method of multipliers*

$$x^{k+1} := \underset{x}{\text{argmin}} \quad L_\rho(x, y^k) \tag{1.7}$$

$$y^{k+1} = y^k + \rho(Ax^{k+1} - b) \tag{1.8}$$

In practice the learning parameter could be anything, instead of ρ, but it is motivated for convergence reasons. The original problem posses the following primal and dual feasibility conditions

$$Ax^* - b = 0, \qquad \nabla f(x^*) + A^T y^* = 0,$$

Now, if we look at the minimization step (1.7) it minimize the augmented Lagrangian,$L_\rho(x, y^k)$

$$
\begin{aligned}
0 &= \nabla_x L(x^{k+1}, y^k) \\
&= \nabla_x f(x) + y^T A + \rho(Ax^{k+1} - b) \\
&= \nabla_x f(x^{k+1}) + A^T y^{k+1}
\end{aligned}
$$

making every step dual feasible. However the method of multipliers has a drawback. Even thought that f is separable, its augmented Lagrangian, L_ρ, is not. This problem is addressed by the Alternating Direction Method of Multipliers

1.4 ADMM algorithm

ADMM mix the decomposability of dual ascent with the superior convergence properties of *method of multipliers.* Given a problem as

$$
\begin{aligned}
&\text{minimize} \quad && f(x) + g(z) \\
&\text{subject to} \quad && Ax + Bz = c
\end{aligned} \tag{1.9}
$$

The associated augmented Lagrangian would be,

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

If we would intended to apply *the method of multipliers* the iteration updates would look like,

$$
\begin{aligned}
(x^{k+1}, z^{k+1}) &:= \underset{x}{\operatorname{argmin}}\, L_\rho(x, z, y^k) \\
y^{k+1} &:= y^k + \rho(Ax^{k+1} + Bz^{k+1} - c)
\end{aligned}
$$

where x and z are jointly optimized. However we an take an additional step and optimize each of the variables separately.

$$
x^{k+1} := \underset{x}{\operatorname{argmin}} \quad L_\rho(x, z^k, y^k) \tag{1.10}
$$

$$
z^{k+1} := \underset{z}{\operatorname{argmin}} \quad L_\rho(x^{k+1}, z, y^k) \tag{1.11}
$$

$$
y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \tag{1.12}
$$

this is *finally* the *Alternating Direction Method of Multipliers.* The *alternating* accounts for the fact that x and z can be minimized in an alternating or sequential fashion.

1.4.1 Scaled form

The ADMM updates can be reformulated in a slightly different form which usually lead to shorter equations. If we define the residual as $r = Ax + Bz - c$ and $u = (1/\rho)y$ as the scaled dual variable, the augmented Lagrangian of the problem (1.9) follows as

$$
L_\rho(x, z, u) = f(x) + g(z) + (\rho/2)||r + u||_2^2 - (\rho/2)||u||_2^2
$$

and then the *scaled* form of ADMM

$$
x^{k+1} := \underset{x}{\operatorname{argmin}} \quad \Big(f(x) + (\rho/2)||Ax + Bz^k - c + u^k||_2^2\Big) \tag{1.13}
$$

$$
z^{k+1} := \underset{z}{\operatorname{argmin}} \quad \Big(g(z) + (\rho/2)||Ax^{k+1} + Bz - c + u^k||_2^2\Big) \tag{1.14}
$$

$$
y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \tag{1.15}
$$

1.4.2 Optimality conditions and Stopping criterion

The optimality conditions for the problem 1.9 are primal feasibility,

$$Ax^* + B^*z - c = 0 \tag{1.16}$$

and dual feasibility,

$$0 \in \partial f(x^*) + A^T y^* \tag{1.17}$$
$$0 \in \partial g(z^*) + B^T y^* \tag{1.18}$$

for the same reasons as explained in the *method of multipliers* the z^{k+1} and y^{k+1} always satisfy (1.18) so we have to only look at (1.17) and (1.16). Since x^{k+1} minimize $L_\rho(x, z^k, y^k)$ by definition we have

$$\begin{aligned} 0 &\in \partial f(x^{k+1}) + A^T y^k + \rho A^T(Ax^{k+1} + Bz^k - c) \\ &= \partial f(x^{k+1}) + A^T(y^k + \rho r^{k+1} \rho B(z^k - z^{k+1})) \\ &= \partial f(x^{k+1}) + A^T y^{k+1} \rho A^T B(z^k - z^{k+1})), \\ \rho A^T B(z^k - z^{k+1}) &= \partial f(x^{k+1}) + A^T y^{k+1}) \end{aligned}$$

So we can define

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

as the *dual residual* for (1.17) and $r^{k+1} = Ax^{k+1} + Bz^{k+1} - c$ as the *primal residual* for (1.16).

2 Inverse Covariance Selection

Zero entries in the inverse covariance matrix correspond to conditional independence of random variables, i.e. knowing the value of one variable do not give information about other knowing the rest. Non-zero entries in the covariance matrix in the case of conditional independence might be contaminated by other variables correlations.

Estimating the inverse covariance for small matrices and fixed sparse patterns is a tractable convex optimization problem. However, when a priori it is not known which variables are conditionally independent it becomes a combinatorial problem which scales exponentially with n. Lasso regularization is an heuristic which address this issue.

2.1 Sparse regularization

Suppose the case where we have samples from a zero mean Gaussian distribution,

$$x_i \sim \mathcal{N}(0, \Sigma), \quad i = 1, 2, ..., N$$

which computational covariance matrix will be denoted as C. One way to estimate the inverse covariance is by means of the Kullback-Leibler divergence which is defined as

$$D_{KL}(\mathcal{N}_1 || \mathcal{N}_0) = \frac{1}{2}\Big(tr(\Sigma_0^{-1}\Sigma_1) + (\mu_0 - \mu_1)^T \Sigma_0^{-1}(\mu_0 - \mu_1) - k + ln\Big(\frac{det\Sigma 0}{det\Sigma_1}\Big)\Big)$$

for two k-dimensional Gaussian distributions. So if we define $S = \Sigma_1^{-1}$ and $X = \Sigma_0$ and minimize the D_{KL} respect to X we have

$$\text{minimize} \quad \text{Tr}(CX) - \text{lndet}X + cte \tag{2.1}$$

after assuming that both distributions have the same mean. Since the Kullback divergence is a measure on how similar two distributions are, after the minimization problem we would have $C = X^{-1}$. Therefore, X is the inverse covariance estimation.

Nonetheless what we are trying to estimate is a sparse general pattern on X which can be introduced through Lasso regularization

$$\text{minimize} \quad \mathrm{Tr}(CX) - \mathrm{lndet}X + \lambda|X|_1 \tag{2.2}$$

2.2 ADMM formulation

The same problem can be expressed as constrained optimization problem with two variables,

$$\begin{array}{ll} \text{minimize} & \mathrm{Tr}(SX) - \mathrm{lndet}X + \lambda|Z|_1 \\ \text{subject to} & X - Z = 0 \end{array} \tag{2.3}$$

now, following the ADMM framework it can be solved iteratively with the updates

$$\begin{aligned} X^{k+1} &:= \underset{x}{\mathrm{argmin}} \quad \Big(\mathrm{Tr}(CX) - \mathrm{lndet}X + (\rho/2)||X - Z^k + U^k||_F^2\Big) \\ Z^{k+1} &:= \underset{z}{\mathrm{argmin}} \quad \Big(\lambda||Z||_1 + (\rho/2)||X - Z^k + U^k||_F^2\Big) \\ U^{k+1} &:= U^k + X^{k+1} + Zk + 1 \end{aligned}$$

the updates can be simplify even further. For example the there exists a closed form solution for the z-minimization step which corresponds to a soft thresholding operation [1, p. 23]

$$Z_{ij}^{k+1} = S_{\lambda/\rho}(X_{ij}^{k+1} + U_{ij}^k)$$

but also the x-minimization step can be expressed as a closed from solution [2, p. 47]

$$X^{k+1} = Q\hat{X}Q^T$$

where Q comes from the orthogonal eigenvalue decomposition of $\rho(Z^k - U^k) - S = Q\Lambda Q^T$ and $\hat{X}$ is a diagonal matrix with the form,

$$\hat{X}_{ii} = \frac{\lambda_i + \sqrt{\lambda_i^2 + 4\rho}}{2\rho}$$

which turns out to be a very cheap computational algorithm, where most of the effort is calculating an eigenvalue decomposition.

2.2.1 Stopping criteria

The algorithm is iterated until the primal and dual residuals

$$R^{k+1} = X - Z \qquad S^{k+1} = \rho(Z^k - Zk + 1)$$

are less than the primal and dual tolerance

$$\epsilon_{primal} > ||R^{k+1}||_2 \qquad \epsilon_{dual} > ||Sk + 1||_2$$

where ϵ_{primal} and ϵ_{dual} are derived from the optimality conditions in the Annex

$$\epsilon_{primal} = \epsilon_{abs}\sqrt{n} + \epsilon_{rel}\max(||X||_2, ||Z||_2) \tag{2.4}$$

$$\epsilon_{dual} = \epsilon_{abs}\sqrt{p} + \epsilon_{rel}||S^T - X^{-T} + \rho U||_2 \tag{2.5}$$

where ϵ_{abs} and ϵ_{rel} are the absolute and relative tolerance for controlling accuracy of the solution.

3 Results

Data from the US senate government [3] has been analyzed in order to find dependencies among politician during voting processes. The dataset consists on all the 2017 senate voting records which has 191 samples of 100 politicians stored in a vector $X \in \mathbb{R}^{100}$, where the first 46 entries are democrat members, the 2 following are independent senators and the rest 52 are republicans.

The computational covariance matrix has been fed to the ADMM algorithm for different regularization problems. In the figure 1 is represented the sparse pattern found for different values of λ, where white space means a zero entry. In one extreme, for $\lambda = 0.3$ we see a diagonal matrix, where each politician vote completely independently. In the other extreme for $\lambda = 0.001$ we see a dense inverse covariance matrix, where everyone depends on everyone else. However, for $\lambda = 0.05$ and 0.01 density is only present in first and fourth quadrant. This results suggest, as expected, that each politician vote depends on what their party fellows are voting.

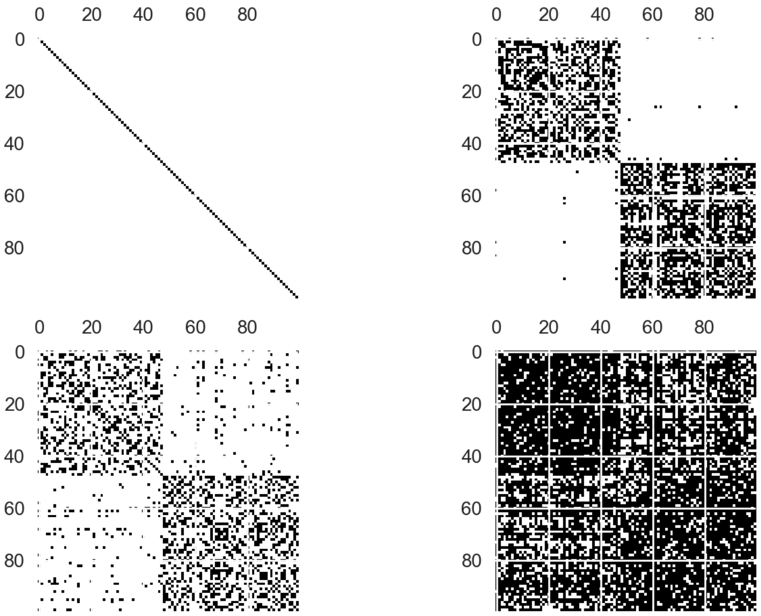

Figure 1: By order, sparse patterns for λ=0.3, 0.05, 0.01 and 0.001

The absolute and relative tolerance in (2.4) and (2.5) for all the experiments have been set to 10^{-3}. The figure 2 shows convergence of primal and dual residuals for different regularization values and fixed ρ. For $\lambda = 0.3$ it takes more time, since the relative tolerance is smaller, while for $\lambda = 0.001$ it takes longer since the residuals decrease slower.

Regularization(λ)	0.3	0.1	0.05	0.01	0.001
Iterations	224	47	40	96	152

Table 1: Number of iterations depending on λ

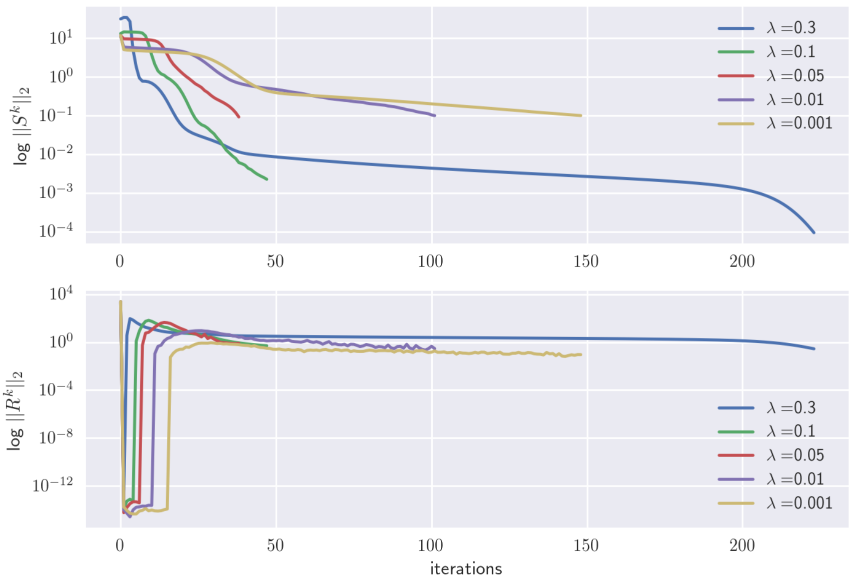

Figure 2: Primal and Dual residuals convergence against λ for fixed ρ

The learning parameter, ρ, does not have impact on the accuracy of the solution but on the convergence speed. High values will decrease the dual function faster but the penalty for breaking primal feasibility will be high. In the figure **??** we can see that the optimal point for $\lambda = 0.02$ is around 0.012.

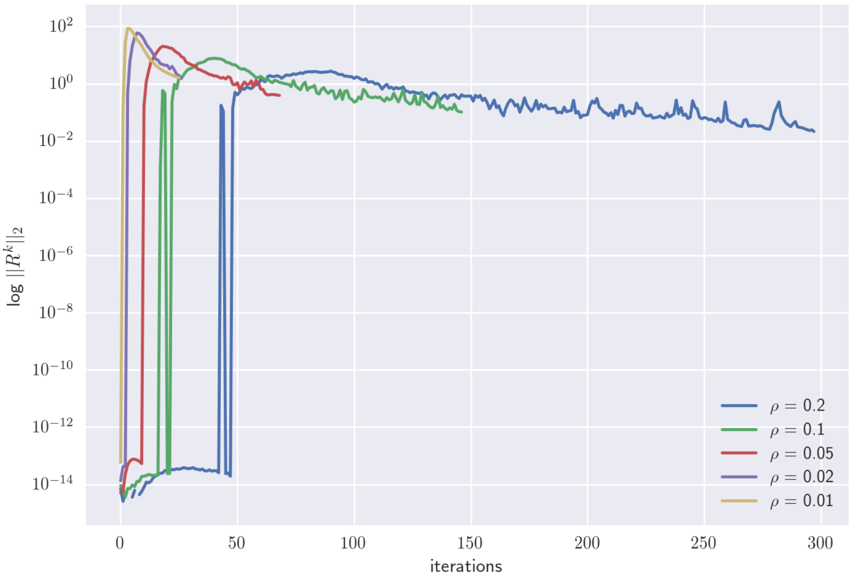

Figure 3: Primal residuals convergence against ρ for $\lambda = 0.02$

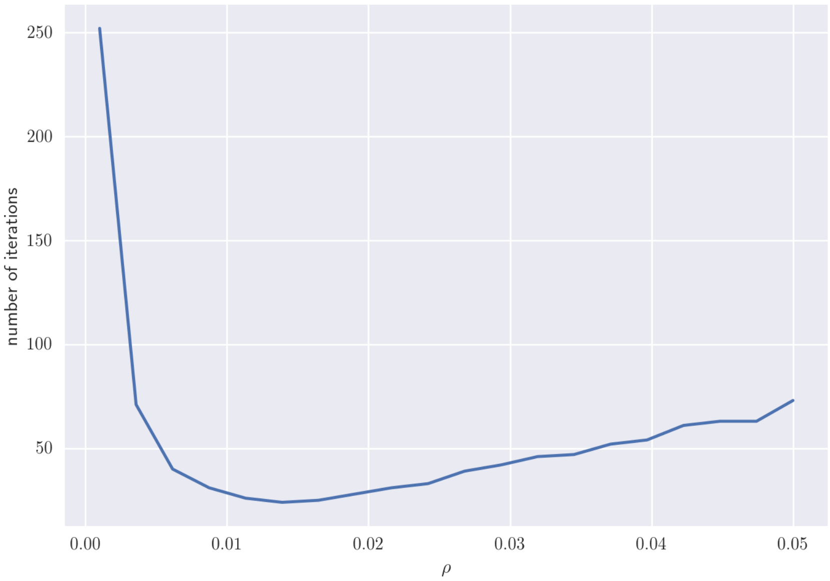

Figure 4: Number of iterations against ρ for $\lambda = 0.02$

4 Annex

Proof of dual tolerance for (2.3). By optimality conditions we know that $\nabla_x L(X, Z^k, U^k) = 0$. Hence,

$$\begin{aligned}
0 &= C^T - (X^{k+1})^{-T} + \rho(X^{k+1} - Z^k + U^k) \\
&= C^T - (X^{k+1})^{-T} + \rho(X^{k+1} - Z^k + U^k) + \rho Z^{k+1} - \rho Z^{k+1} \\
&= C^T - (X^{k+1})^{-T} + \rho(R^{k+1} + U^k) + \rho(Z^{k+1} - Z^k) \\
&= C^T - (X^{k+1})^{-T} + \rho U^{k+1} + \rho(Z^{k+1} - Z^k) \\
\rho(Z^{k+1} - Z^k) &= C^T - (X^{k+1})^{-T} + \rho U^{k+1} \\
\rho S^{k+1} &= C^T - (X^{k+1})^{-T} + \rho U^{k+1}
\end{aligned}$$

which demonstrate the choice of (2.5).

References

[1] R. T. Rockafellar. *Convex Analysis.* 1970.

[2] E. Chu B. Peleato S. Boyd, N. Parikh and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 2010.

[3] govTrack. Voting records.