# ADMM Algorithm for Graphical Lasso with an $\ell_{\infty}$ Element-wise Norm Constraint

#### Karthik Mohan

December 2, 2013

#### Abstract

We consider the problem of Graphical lasso with an additional  $\ell_{\infty}$  element-wise norm constraint on the precision matrix. This problem has applications in high-dimensional covariance decomposition such as in [Janzamin and Anandkumar, 2012a]. We propose an ADMM algorithm to solve this problem. We also use a continuation strategy on the penalty parameter to have a fast implementation of the algorithm.

## 1 Problem

The graphical lasso formulation with  $\ell_{\infty}$  element-wise norm constraint is as follows:

$$\min_{\mathbf{\Theta} \in \mathbb{R}^{p \times p}, \mathbf{\Theta} \succ 0} -\log \det(\mathbf{\Theta}) + \langle \mathbf{S}, \mathbf{\Theta} \rangle + \gamma \|\mathbf{\Theta} - \operatorname{diag}(\mathbf{\Theta})\|_{1}$$
s.t. 
$$\|\mathbf{\Theta} - \operatorname{diag}(\mathbf{\Theta})\|_{\infty} \le \lambda,$$
(1)

where  $\|\cdot\|_1$  denotes the  $\ell_1$  norm, and  $\|\cdot\|_{\infty}$  denotes the  $\ell_{\infty}$  element-wise norm of a matrix. For a matrix X,  $\|X\|_{\infty} = \max_{i,j} |X_{ij}|$ . This formulation first appeared in [Janzamin and Anandkumar, 2012a] in the context of high-dimensional covariance decomposition. We next provide an efficient ADMM algorithm to solve (1).

## 2 ADMM approach

The alternating direction method of multipliers (ADMM) algorithm [Boyd et al., 2011, Eckstein, 2012] is especially suited to solve optimization problems whose objective can be decomposed into the sum of many *simple* convex functions. By simple, we mean a function whose proximal operator can be computed efficiently. The proximal operator of a function f is given by:

$$\operatorname{Prox}_{f}(\boldsymbol{A}, \lambda) = \operatorname{argmin}_{X} \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{A}\|_{F}^{2} + \lambda f(\boldsymbol{X})$$
 (2)

Consider the following optimization problem:

$$\min_{\boldsymbol{X},\boldsymbol{Y}} \quad f(\boldsymbol{X}) + g(\boldsymbol{Y}) 
\text{s.t.} \quad \boldsymbol{X} = \boldsymbol{Y},$$
(3)

where we assume that f and g are simple functions.

The ADMM algorithm alternatively optimizes the augmented Lagrangian to (3), which is given by:

$$\mathcal{L}_{\rho}(\boldsymbol{X}, \boldsymbol{Y}, \Lambda) = f(\boldsymbol{X}) + g(\boldsymbol{Y}) + \langle \Lambda, \boldsymbol{X} - \boldsymbol{Y} \rangle + \frac{\rho}{2} \|\boldsymbol{X} - \boldsymbol{Y}\|_F^2.$$
(4)

The (k+1)th iteration of ADMM is then given by:

$$X^{k+1} \leftarrow \underset{\boldsymbol{X}}{\operatorname{argmin}} \mathcal{L}_{\rho}(\boldsymbol{X}, \boldsymbol{Y}^{k}, \Lambda^{k})$$

$$Y^{k+1} \leftarrow \underset{\boldsymbol{Y}}{\operatorname{argmin}} \mathcal{L}_{\rho}(X^{k+1}, \boldsymbol{Y}, \Lambda^{k})$$

$$\Lambda^{k+1} \leftarrow \Lambda^{k} + \rho(X^{k+1} - \boldsymbol{Y}^{k+1})$$
(5)

Note that each iteration in (5) has closed form updates if f and g have closed form proximal operators. The ADMM algorithm has a  $O(1/\epsilon)$  convergence rate [Goldfarb et al., 2010] just as for proximal gradient descent. We next reformulate our problem (1) and construct an ADMM algorithm.

## 3 Reformulation by introducing new variables

We now reformulate (1) into the standard form in (3) to derive an ADMM algorithm for our problem. We first define some notation. Let,

$$X = \begin{bmatrix} \mathbf{\Theta} \\ \Gamma \end{bmatrix} \in \mathbb{R}^{2p \times p}$$

$$Y = \begin{bmatrix} \hat{\mathbf{\Theta}} \\ \hat{\Gamma} \end{bmatrix} \in \mathbb{R}^{2p \times p}$$

$$f(X) = -\log \det(\mathbf{\Theta}) + \langle S, \mathbf{\Theta} \rangle + \mathcal{I}_{\{\mathbf{\Theta} \succ 0\}} + \gamma \| \Gamma - \operatorname{diag}(\Gamma) \|_{1}$$

$$g(Y) = \mathcal{I}_{\{\|\hat{\mathbf{\Theta}} - \operatorname{diag}(\hat{\mathbf{\Theta}})\|_{\infty} \le \lambda\}} + \mathcal{I}_{\{\hat{\mathbf{\Theta}} = \hat{\Gamma}\}},$$

$$(6)$$

where  $\mathcal{I}_{\{.\}}$  denotes the indicator function that equals zero if the statement inside  $\{.\}$  is true and  $\infty$  otherwise. Then note that (1) is equivalent to (3) with X, Y, f, g as in (6).

## 4 ADMM algorithm for Glasso with an $\ell_{\infty}$ element-wise norm constraint

Define the following operators:

Expand
$$(\boldsymbol{A}; \rho) = \underset{\boldsymbol{\Theta}}{\operatorname{argmin}} - \log \det(\boldsymbol{\Theta}) + \frac{\rho}{2} \|\boldsymbol{\Theta} - \boldsymbol{A}\|_{F}^{2}$$

$$S(\boldsymbol{A}; \gamma) = \underset{\boldsymbol{\Gamma}}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{\Gamma} - \boldsymbol{A}\|_{F}^{2} + \gamma \|\boldsymbol{\Gamma} - \operatorname{diag}(\boldsymbol{\Gamma})\|_{1}$$

$$\mathcal{P}_{\infty}(\boldsymbol{A}; \lambda) = \underset{\boldsymbol{\Theta}: \|\widetilde{\boldsymbol{\Theta}} - \operatorname{diag}(\widetilde{\boldsymbol{\Theta}})\|_{\infty} \leq \lambda}{\operatorname{argmin}} \|\widetilde{\boldsymbol{\Theta}} - \boldsymbol{A}\|_{F}^{2}$$

$$(7)$$

Plugging in the choice of X, Y and f(X), g(Y) from (6) into (5), we get the following algorithm:

```
input: \rho > 0;
Initialize: Primal variables to the identity matrix and dual variables to the zero matrix;
while Not\ converged\ \mathbf{do}
\Theta \leftarrow \operatorname{Expand}\left(\hat{\mathbf{\Theta}} - (S + \Lambda_{\mathbf{\Theta}})/\rho; \rho\right);
\Gamma \leftarrow \mathcal{S}\left(\hat{\Gamma} - \Lambda_{\Gamma}/\rho; \frac{\gamma}{\rho}\right);
\hat{\mathbf{\Theta}} \leftarrow \mathcal{P}_{\infty}\left(\frac{1}{2}(\mathbf{\Theta} + \Gamma) + \frac{(\Lambda_{\mathbf{\Theta}} + \Lambda_{\Gamma})}{2\rho}; \lambda\right);
\hat{\Gamma} = \hat{\mathbf{\Theta}};
\Lambda_{\mathbf{\Theta}} = \Lambda_{\mathbf{\Theta}} + \rho(\mathbf{\Theta} - \hat{\mathbf{\Theta}});
\Lambda_{\Gamma} = \Lambda_{\Gamma} + \rho(\Gamma - \hat{\Gamma})
end
```

**Algorithm 1:** ADMM algorithm for graphical lasso with an additional  $\ell_{\infty}$  norm constraint in (1)

Note that we have

Expand
$$(\mathbf{A}; \rho) = \frac{\rho A + (\rho^2 A^2 + 4\rho I)^{1/2}}{2\rho}$$
  

$$\mathcal{S}(\mathbf{A}; \gamma) = \text{Soft-Threshold}_{\gamma}(A)$$

$$\mathcal{P}_{\infty}(\mathbf{A}; \lambda) = \text{Clip}_{\lambda}(A),$$

where

Soft-Threshold<sub>\gamma</sub>(A)<sub>ij</sub> = 
$$\begin{cases} A_{ij} & i = j \\ \operatorname{sign}(A_{ij}) \max(|A_{ij}| - \gamma, 0) & i \neq j \end{cases}$$

and

$$\operatorname{Clip}_{\lambda}(A)_{ij} = \begin{cases} A_{ij} & i = j \\ \operatorname{sign}(A_{ij}) \min(|A_{ij}|, \lambda) & i \neq j. \end{cases}$$

We add a continuation scheme to Algorithm 1, where  $\rho$  is varied through the iterations. Specifically, we initially set  $\rho = 1$  and double  $\rho$  every 20 iterations. We terminate the algorithm when the relative error is small or when  $\rho$  is too large as

$$\frac{\|\Lambda_{\boldsymbol{\Theta}}^{k+1} - \Lambda_{\boldsymbol{\Theta}}^k\|_F}{\max(1, \|\Lambda_{\boldsymbol{\Theta}}^k\|_F)} < \epsilon \quad \text{or} \quad \rho > 10^6.$$

We apply our algorithm to the problem of high-dimensional covariance decomposition, details of which can be found in [Janzamin and Anandkumar, 2012b].

### References

- M. Janzamin and A. Anandkumar. High-dimensional covariance decomposition into sparse markov and independence domains. In Proceedings of the 29th International Conference on Machine Learning, 2012a.
- S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and trends in machine learning*, 3(1): 1–122, 2011.
- J. Eckstein. Augmented lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results. 2012. RUTCOR research report RRR 32-2012.
- D. Goldfarb, S. Ma, and K. Scheinberg. Fast alternating linearization methods for minimizing the sum of two convex functions. 2010. Submitted to SIAM J. on optimization.
- M. Janzamin and A. Anandkumar. High-dimensional covariance decomposition into sparse markov and independence models. *Arxiv preprint* arXiv:1211.0919 [stat.ML], 2012b.