Optimization Methods in Sparse Approximation

With Applications to Basis Pursuit and Gaussian Graphical Models

Syed Rahman

Introduction: The goal of this project is to look at various optimization algorithms that solve the noisy basis pursuit ℓ_1 optimization problem as follows:

$$\begin{aligned} &\operatorname*{argmin}_{\beta} F(\beta) \\ =&\operatorname*{argmin}_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \end{aligned} \tag{1}$$

The algorithms we will be looking at are the ISTA, FISTA, ADMM. The basic idea is to look at all these methods in details. This will include running our own simulations to compare times for each of these, study the effect of step-sizes and discuss convergence issues/properties wherever possible. In addition, we also adapt these methods to gaussian graphical models where the basic problem can be stated as

$$\begin{aligned} &\operatorname*{argmin}_{\Omega \succ 0} \ell(\Omega) + \lambda\|\Omega\|_1 \\ =&\operatorname*{argmin}_{\Omega \succ 0} \operatorname{trace}(\Omega S) - \log|\Omega| + \lambda\|\Omega\|_1 \end{aligned}$$

Subgradent Methods: Note that in sub-gradient descent we have the basic update

$$\beta^k = \beta^{k-1} - t_k g^{k-1},$$

where t_k is the step-size and g^{k-1} is the sub-gradient. For our problem, the subgradient is $-X^t(y - X\beta) + \lambda s$ where

$$s_i = \begin{cases} \operatorname{sign}(\beta_i) & \text{if } \beta_i \neq 0 \\ [-1, 1] & \text{if } \beta_i = 0 \end{cases}$$

For the subgradient, we will just use

$$s_i = \begin{cases} \operatorname{sign}(\beta_i) & \text{if } \beta_i \neq 0 \\ 0 & \text{if } \beta_i = 0 \end{cases}$$

For back-tracking line search, fix $\eta \in (0,1)$. At each iteration, while

$$F(\beta - t\partial F(\beta)) > F(\beta) - \frac{t}{2}\|\partial F(\beta)\|^2$$

let $t = \eta t$. Hence the goal is to find the smallest i s.t.

$$F(\beta - t\partial F(\beta)) < F(\beta) - \frac{\eta^i t}{2}\|\partial F(\beta)\|^2$$

Here $F(\beta) = \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$ and $\partial F(\beta) = -X^t(y - X\beta) + \lambda s$

Algorithm 1 Subgradeint Algorithm with fixed step size

Set $\epsilon \in \mathbb{R}$
Set $\beta^{old} \in \mathbb{R}^p$
Set $t \in \mathbb{R}$
Set $\beta^{new} \leftarrow \beta^{old} - t(-X^t(y - X\beta^{old}) + \lambda s)$ where

$$s_i = \begin{cases} \text{sign}(\beta_i^{old}) & \text{if } \beta_i \neq 0 \\ 0 & \text{if } \beta_i^{old} = 0 \end{cases}$$

while $\left|\left|\beta^{new} - \beta^{old}\right|\right|_\infty \geqslant \epsilon$ **do**
  Set $\beta^{new} \leftarrow \beta^{old} - t(-X^t(y - X\beta^{old}) + \lambda s)$
end while

Algorithm 2 Subgradeint Algorithm with diminishing step size

Set $\epsilon \in \mathbb{R}$
Set $\eta \in \mathbb{R}(0,1)$
Set $\beta^{old} \in \mathbb{R}^p$
Set $t = 1$
while $F(\beta - \partial F(\beta)) > F(\beta) - \frac{t}{2}\|\partial F(\beta)\|^2$ **do**
  Set $t = \eta t$
end while
Set $\beta^{new} \leftarrow \beta^{old} - t(-X^t(y - X\beta^{old}) + \lambda s)$ where

$$s_i = \begin{cases} \text{sign}(\beta_i^{old}) & \text{if } \beta_i \neq 0 \\ 0 & \text{if } \beta_i^{old} = 0 \end{cases}$$

while $\left|\left|\beta^{new} - \beta^{old}\right|\right|_\infty \geqslant \epsilon$ **do**
  while $F(\beta - \partial F(\beta)) > F(\beta) - \frac{t}{2}\|\partial F(\beta)\|^2$ **do**
    Set $t = \eta t$
  end while
  Set $\beta^{new} \leftarrow \beta^{old} - t(-X^t(y - X\beta^{old}) + \lambda s)$
end while

Theorem 1. For fixed step sizes, the sudgradient method satisfies

$$\lim_{k\to\infty} F(\beta^k) \leqslant F(\beta^*) + \frac{L^2 t}{2}$$

with convergenve rate of $O(\frac{1}{\sqrt{k}})$, where $|F(\beta^1) - F(\beta^2)| \leqslant L\|\beta^1 - \beta^2\|$.
Theorem 2. For diminshing step sizes, the sudgradient method satisfies

$$\lim_{k\to\infty} F(\beta^k) = F(\beta^*)$$

with convergenve rate of $O(\frac{1}{\sqrt{k}})$.

For the covariance estimation problem, the subgradient is $S - \Omega^{-1} + \lambda\Gamma$ with

$$\Gamma_{ij} = \begin{cases} \text{sign}(\Omega_{ij}) & \text{if } \Omega_{ij, i\neq j} \neq 0 \\ 0 & \text{if } \Omega_{ij} = 0, \Omega_{ij, i=j} \end{cases}$$

Hence the basic update is:

$$\Omega^k = \Omega^{k-1} - t_k(S - (\Omega^{k-1})^{-1} + \lambda\Gamma^{k-1})$$

ISTA/FISTA For this part, we look at *A Fast Iterative Shrinkage Thresholding Algorithm for Linear Inverse Problems* by Amir Beck and Marc Teboulle. These are proximal gradient methods. The proximal operator for the ℓ_1 penalty, $h(\beta) = \lambda \|\beta\|_1$ is

$$\begin{aligned}
\text{prox}_t(\beta) &= \underset{\eta}{\text{argmin}} \frac{1}{2t} \|\beta - \eta\|_2^2 + h(\eta) \\
&= \underset{\eta}{\text{argmin}} \frac{1}{2t} \|\beta - \eta\|_2^2 + \lambda \|\eta\|_1 \\
&= S_{\lambda t}(\beta)
\end{aligned}$$

where $[S_{\lambda t}(x)]_i = \text{sign}(x_i) * \max\{|x_i| - \lambda t, 0\}$. In general, if we want to minimize $F(\beta) = g(\beta) + h(\beta)$, we do:

$$\beta^{(k)} = \text{prox}_{t_k h}(\beta^{(k-1)} - t_k \nabla g(\beta^{(k-1)}))$$

To see why this works, note that

$$\begin{aligned}
\beta^+ &= \underset{\eta}{\text{argmin}}(h(\eta) + \frac{1}{2t} \|\eta - \beta + t\nabla g(\beta)\|_2^2) \\
&= ... \\
&= \underset{\eta}{\text{argmin}}(h(\eta) + g(\beta) + \nabla g(\beta)^t(\eta - \beta) + \frac{1}{2t} \|\eta - \beta\|_2^2)
\end{aligned}$$

Hence, we are essentially minimizing $h(\eta)$ plus a simple local model of $g(\eta)$ around β. Recall, the 2^{nd} order Taylor series approximation to $g(\eta)$ near β is

$$\begin{aligned}
g(\eta) =& g(\beta) + \nabla g(\beta)^t(\eta - \beta) + (\eta - \beta)^t \nabla^2 g(\beta)(\eta - \beta) \\
&\leqslant g(\beta) + \nabla g(\beta)^t(\eta - \beta) + L(\eta - \beta)^t(\eta - \beta)
\end{aligned}$$

where the function $\nabla g(\beta)$ has Lipschitz constant L. Also note that the lasso objective function can be rewritten as $g(\beta) + h(\beta)$ here h is as before and $g(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$. Then $\nabla g(\beta) = -X^t(y - X\beta)$. Then the update for ISTA is as follows:

$$\beta^k = S_{\lambda t}(\beta^{k-1} + tX^t(y - X\beta^{k-1}))$$

In 1983, Nesterov proposed the following Accelarated gradient descent algorithm for convex, differentiable functions $g(\beta)$:

$$\beta^{k+1} = \eta^k - t_k \nabla g(\eta^k)$$
$$\eta^{k+1} = (1-\gamma_k)\beta^{k+1} + \gamma_k \beta^k$$

with convergence rate $O(\frac{1}{k^2})$. FISTA is essentially a combination of this with proximal gradient methods. The update is as follows:

$$t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$$
$$\gamma = \beta^{k-1} + \frac{t_k - 1}{t_{k+1}}(\beta^{k-1} - \beta^{k-2})$$
$$\beta^k = S_{\lambda t}(\gamma + tX^t(y - X\gamma))$$

Finally, we will discuss convergence properties for the back-tracking line search. Note that in this case we know the Lipschitz constant to be $\lambda_{max}(X^tX)$, i.e.

$$\begin{aligned}\|\nabla g(\beta_1) - \nabla g(\beta_2)\|_2 &\leqslant L \|\beta_1 - \beta_2\|_2 \\ &= \lambda_{max}(X^tX) \|\beta_1 - \beta_2\|_2\end{aligned}$$

Hence we can take $t = \frac{1}{\lambda_{max}(X^tX)}$.

Algorithm 3 ISTA with fixed step size

Set $t \leftarrow \frac{1}{\lambda_{max}(X^tX)}$
Set $\epsilon \in \mathbb{R}$
Set $\beta^{old} \in \mathbb{R}^p$
Set $\beta^{new} \leftarrow S_{\lambda t}(\beta^{old} + tX^t(y - X\beta^{old}))$
while $\left|\left|\beta^{new} - \beta^{old}\right|\right|_\infty \geqslant \epsilon$ **do**
    $\beta^{new} \leftarrow S_{\lambda t}(\beta^{old} + tX^t(y - X\beta^{old}))$
end while

If we didn't know the step-size we can use back-tracking line search. For this let

$$F(\beta) = g(\beta) + h(\beta) = \frac{1}{2}\|y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

and

$$Q_L(\beta^{new}, \beta^{old}) = g(\beta^{old}) + < \beta^{new} - \beta^{old}, \nabla g(\beta^{old}) > + \frac{L}{2}\left|\left|\beta^{new} - \beta^{old}\right|\right|_2^2 + h(\beta^{new})$$

Now if $F(\beta^{new}) > Q_L(\beta^{old}, \beta^{new})$, set $t = \beta t$ for some $\beta < 1$. Note that,

$$\begin{aligned} & F(\beta^{new}) > Q_L(\beta^{old}, \beta^{new}) \\ \iff & g(\beta^{new}) + h(\beta^{new}) > g(\beta^{old}) + < \beta^{new} - \beta^{old}, \nabla g(\beta^{old}) > \\ & \quad + \frac{L}{2} \left|\left|\beta^{new} - \beta^{old}\right|\right|_2^2 + h(\beta^{new}) \\ \iff & g(\beta^{new}) > g(\beta^{old}) + < \beta^{new} - \beta^{old}, \nabla g(\beta^{old}) > \\ & \quad + \frac{L}{2} \left|\left|\beta^{new} - \beta^{old}\right|\right|_2^2 \end{aligned}$$

which may be a slightly less computationally intensive way to pick the step-size.

Algorithm 4 ISTA with diminishing step size

Set $\epsilon \in \mathbb{R}$
Set $\eta \in \mathbb{R}(0, 1)$
Set $\beta^{old} \in \mathbb{R}^p$
Set $L^{old} > 0$
Set $t = \frac{1}{L^{old}}$
Find smallest integer i such that $F(\beta^{new}) \leqslant Q_{L^{new}}(\beta^{new}, \beta^{old})$ with $\frac{1}{L^{new}} = \eta^i \frac{1}{L^{old}}$ and $t = \frac{1}{L^{new}}$
Set $\beta^{new} \leftarrow S_{\lambda t}(\beta^{old} + tX^t(y - X\beta^{old}))$
while $\left|\left|\beta^{new} - \beta^{old}\right|\right|_\infty \geqslant \epsilon$ **do**
    Find smallest integer i such that $F(\beta^{new}) \leqslant Q_{L^{new}}(\beta^{new}, \beta^{old})$ with $\frac{1}{L^{new}} = \eta^i \frac{1}{L^{old}}$ and $t = \frac{1}{L^{new}}$
    $\beta^{new} \leftarrow S_{\lambda t}(\beta^{old} + tX^t(y - X\beta^{old}))$
end while

Theorem 3. Let β^k be a sequence generated by either of the ISTA algorithms as described above. Then for any $k \geqslant 1$

$$F(\beta_k) - F(\beta^*) \leqslant \frac{\alpha L(g) \left\|\beta_0 - \beta^*\right\|_2}{2k}$$

where $\alpha = 1$ for constant step size and $\alpha = \eta$ for back-tracking line search.

Algorithm 5 FISTA with fixed step size

Set $t \leftarrow \frac{1}{\lambda_{max}(X^t X)}$
Set $t_1 \leftarrow 1$
Set $\epsilon \in \mathbb{R}$
Set $k \leftarrow 1$
Set $\beta^{old} \leftarrow \zeta^0 \in \mathbb{R}^p$
Set $\beta^{new} \leftarrow S_{\lambda t}(\zeta^0 + tX^t(y - X\zeta^0))$
$t_{k+1} \leftarrow \frac{1+\sqrt{1+4t_k^2}}{2}$
$\zeta^0 \leftarrow \beta^{new} + \frac{t_k - 1}{t_{k+1}}(\beta^{new} - \beta^{old})$
while $\left|\left|\beta^{new} - \beta^{old}\right|\right|_\infty \geqslant \epsilon$ **do**
    $\beta^{old} \leftarrow \beta^{new}$
    $\beta^{new} \leftarrow S_{\lambda t}(\zeta^0 + tX^t(y - X\zeta^0))$
    $t_{k+1} \leftarrow \frac{1+\sqrt{1+4t_k^2}}{2}$
    $\zeta^0 \leftarrow \beta^{new} + \frac{t_k - 1}{t_{k+1}}(\beta^{new} - \beta^{old})$
    $k \leftarrow k + 1$
end while

Algorithm 6 FISTA with diminishing step size

Set $t_1 \leftarrow 1$
Set $\epsilon \in \mathbb{R}$
Set $k \leftarrow 1$
Set $\beta^{old} \leftarrow \zeta^0 \in \mathbb{R}^p$
Find smallest integer i such that $F(\beta^{new}) \leqslant Q_{L^{new}}(\beta^{new}, \beta^{old})$ with $\frac{1}{L^{new}} = \eta^i \frac{1}{L^{old}}$ and $t = \frac{1}{L^{new}}$
Set $\beta^{new} \leftarrow S_{\lambda t}(\zeta^0 + tX^t(y - X\zeta^0))$
$t_{k+1} \leftarrow \frac{1+\sqrt{1+4t_k^2}}{2}$
$\zeta^0 \leftarrow \beta^{new} + \frac{t_k - 1}{t_{k+1}}(\beta^{new} - \beta^{old})$
while $\left|\left|\beta^{new} - \beta^{old}\right|\right|_\infty \geqslant \epsilon$ **do**
  $\beta^{old} \leftarrow \beta^{new}$
  Find smallest integer i such that $F(\beta^{new}) \leqslant Q_{L^{new}}(\beta^{new}, \beta^{old})$ with $\frac{1}{L^{new}} = \eta^i \frac{1}{L^{old}}$ and $t = \frac{1}{L^{new}}$
  $\beta^{new} \leftarrow S_{\lambda t}(\zeta^0 + tX^t(y - X\zeta^0))$
  $t_{k+1} \leftarrow \frac{1+\sqrt{1+4t_k^2}}{2}$
  $\zeta^0 \leftarrow \beta^{new} + \frac{t_k - 1}{t_{k+1}}(\beta^{new} - \beta^{old})$
  $k \leftarrow k + 1$
end while

Theorem 4. Let β^k be a sequence generated by either of the FISTA algorithms as described above. Then for any $k \geqslant 1$

$$F(\beta_k) - F(\beta^*) \leqslant \frac{\alpha L(g) \left\|\beta_0 - \beta^*\right\|_2}{(k+1)^2}$$

where $\alpha = 1$ for constant step size and $\alpha = \eta$ for back-tracking line search.

To adapt this for the `glasso` problem as discussed last week, note that we want to minimize

$$\begin{aligned} &\operatorname{trace}(\Omega S) - \log|\Omega| + \lambda \left\|\Omega\right\|_1 \\ =&\ell(\Omega) + \lambda \left\|\Omega\right\|_1 \end{aligned}$$

Now, $\nabla \ell(\Omega) = S - \Omega^{-1}$. Hence the ISTA update would be

$$\Omega^{new} = S_{\lambda t}(\Omega^{old} + t(S - (\Omega^{old})^{-1}))$$

Similarly, FISTA would be

$$\begin{aligned} t_{k+1} &= \frac{1+\sqrt{1+4t_k^2}}{2} \\ \zeta^{old} &= \Omega^{new} + \frac{t_k - 1}{t_{k+1}}(\Omega^{new} - \Omega^{old}) \\ \Omega^{new} &= S_{\lambda t}(\zeta^{old} + t(S - (\zeta^{old})^{-1})) \end{aligned}$$

In both the above formulations the soft-thresholding is only applied to the off-diagonal elements of the matrices.

ADMM For this part, we refer to *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers* by Stephen Boyd, Neal Parikh, Eric Chu Borja Peleato and Jonathan Eckstein. The ADMM algorithm mixes the decomposability of the *dual ascent method* with the superior convergence properties of the *method of multipliers*. Note that we can restate Equation 1:

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

as:

$$\frac{1}{2}\left\|y - X\beta\right\|_2^2 + \lambda\left\|\gamma\right\|_1 + \frac{\rho}{2}\left\|\beta - \gamma\right\|_2^2 \text{ s.t. } \beta = \gamma$$

The augmented Lagrangian in such as case is:

$$L(\beta, \gamma, \eta) = \frac{1}{2}\left\|y - X\beta\right\|_2^2 + \lambda\left\|\gamma\right\|_1 + \frac{\rho}{2}\left\|\beta - \gamma\right\|_2^2 + \eta^t(\beta - \gamma)$$

The ADMM updates in this case are:

$$\begin{aligned}
\beta^k &= \underset{\beta}{\operatorname{argmin}}\, L(\beta^{k-1}, \gamma, \eta) \\
\gamma^k &= \underset{\gamma}{\operatorname{argmin}}\, L(\beta, \gamma^{k-1}, \eta) \\
\eta^k &= \eta^{k-1} + \rho(\beta - \gamma)
\end{aligned}$$

Step 3 is trivial. For step 1, we simply calculate the derivative and set it to 0.

$$\begin{aligned}
\nabla_\beta L(\beta, \gamma, \eta) = -X^t(y - X\beta) + \rho(\beta - \gamma) + \eta &\overset{set}{=} 0 \\
\iff X^t(y - X\beta) - \rho\beta &= -\rho\gamma + \eta \\
\iff +X^tX\beta + \rho\beta &= +\rho\gamma - \eta + X^ty \\
\iff \beta &= (X^tX + \rho I)^{-1}(\rho\gamma - \eta + X^ty)
\end{aligned}$$

Finally for step 2,

$$\partial_\gamma L(\beta, \gamma, \eta) = \lambda s + \rho(\gamma - \beta) - \eta$$

where

$$s_i = \begin{cases} 1 & \text{if } \gamma_i > 0 \\ -1 & \text{if } \gamma_i < 0 \\ [-1, 1] & \text{if } \gamma_i = 0 \end{cases}$$

Thus, $\gamma = S_{\frac{\lambda}{\rho}}(\beta + \frac{\eta}{\rho})$.

Algorithm 7 ADMM

Set $\epsilon \in \mathbb{R}$
Set $\rho \in \mathbb{R}_+$
$\beta^{old} \leftarrow \gamma^{old} \leftarrow \eta^{old} \in \mathbb{R}^p$
$\beta^{new} \leftarrow (X^tX + \rho I)^{-1}(\rho\gamma^{old} - \eta^{old} + X^ty)$
$\gamma^{new} = S_{\frac{\lambda}{\rho}}(\beta^{new} + \frac{\eta^{old}}{\rho})$
$\eta^{new} = \eta^{old} + \rho(\beta^{new} - \gamma^{new})$
while $\left|\left|\beta^{new} - \beta^{old}\right|\right|_\infty \geqslant \epsilon$ **do**
    $\beta^{old} \leftarrow \beta^{new}$
    $\gamma^{old} \leftarrow \gamma^{new}$
    $\eta^{old} \leftarrow \eta^{new}$
    $\beta^{new} \leftarrow (X^tX + \rho I)^{-1}(\rho\gamma^{old} - \eta^{old} + X^ty)$
    $\gamma^{new} \leftarrow S_{\frac{\lambda}{\rho}}(\beta^{new} + \frac{\eta^{old}}{\rho})$
    $\eta^{new} \leftarrow \eta^{old} + \rho(\beta^{new} - \gamma^{new})$
end while

Now to apply this to sparse inverse covariance selection recall that we want so solve

$$\min_{\Omega} \operatorname{trace}(S\Omega) - \log|\Omega| + \lambda\left|\left|\Omega\right|\right|_1$$

which is equivalent to

$$\min_{\Omega, Z} \operatorname{trace}(S\Omega) - \log|\Omega| + \lambda\left|\left|Z\right|\right|_1 \text{ s.t } \Omega = Z$$

The augmented Lagrangian in such a case is:

$$L(\Omega, Z, Y) = \operatorname{trace}(S\Omega) - \log|\Omega| + \lambda\left|\left|Z\right|\right|_1 + Y^t(\Omega - Z) + \frac{\rho}{2}\left|\left|\Omega - Z\right|\right|_F^2$$

This can be decomposed to 3 separate optimization problems as follows:

$$\min_{\Omega} \operatorname{trace}(S\Omega) - \log|\Omega| + Y^t(\Omega - Z) + \frac{\rho}{2}\|\Omega - Z\|_F^2$$
$$\min_{Z} \lambda\|Z\|_1 + Y^t(\Omega - Z) + \frac{\rho}{2}\|\Omega - Z\|_F^2$$
$$\max_{Y} Y^t(\Omega - Z)$$

Numerical Experiments for BP: For the basis pursuit problem, we generated data in the following manner. We set $p = 200$ and $n = \{20, 50, 100, 500\}$. The number of non-zero elements of β^* was set equal to 20. $X_{ij} \overset{iid}{\sim} \mathcal{N}(0,1); i = 1, ..., n; j = 1, ..., p$, $E_i \overset{iid}{\sim} \mathcal{N}(0,1); i = 1, ..., n$ and $y = X\beta^* + E$. Finally, λ was picked through 5-fold cross-validation.

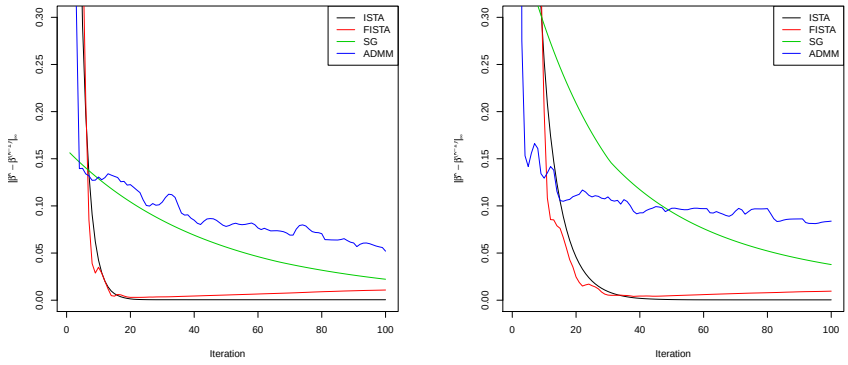
Figure 1 shows $\left|\left|\beta^k - \beta^{k-1}\right|\right|_\infty$ at each step for the methods. It is clear that according to this measure, ISTA and FISTA perform the best for $n < p$, while ADMM performs the best for the $n > p$ case. Figure 3 shows the timing comaprisons while Figure 2 displays $\frac{\left|\left|\hat{\beta} - \beta^*\right|\right|_2}{||\beta^*||_2}$ for all the methods. At $n = 20$, the performance of ADMM is odd because it starts to rise after initially declining. This is not the case for all the other values of n, where it clearly outperforms the other methods. In the above case, X was a well conditioned matrix. We repeated the experiments with an ill-conditioned X. The performance of the subgradient method was very poor showing how this algorithm lacks stability. ISTA/FISTA's performance was pretty good, but inconsistent. The most reilable was the ADMM algorithm, whose performance hardly changed. These results are shown in Figures 5,4 and 6.

Numerical Experiments for GGM: We set $p = 500$ and $n = 1000$. Approximately 95% of the entries in Ω^* were set to 0. We generate $X_i \overset{iid}{\sim} \mathcal{N}_p(0, \Omega^{-1})$ for $i = 1, ..., n$. Let X_i be the i^{th} row of X. Set $S = \frac{1}{n}X^tX$. We compared $\frac{\left|\left|\hat{\Omega} - \Omega^*\right|\right|_F}{||\Omega^*||_F}$ for all the methods: SG(0.487), ISTA(0.463), FISTA(0.463), ADMM(0.553). These were all inferior to GLASSO(0.346) discussed last week in class. However, we are more interested in recovering the sparsity pattern here. Table 1 shows the true positive and false positive rates. According to these, while ISTA and FISTA have a higher true recovery rate, they also have a higher error rate. The subgradient method recovers a very dense graph, while ADMM recovers a very sparse one.

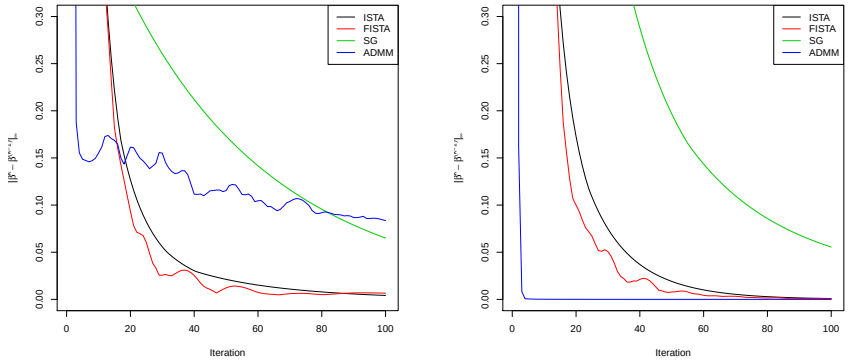
	Method	TPR	FPR
1	SG	1.000	0.914
2	ISTA	0.728	0.841
3	FISTA	0.728	0.841
4	ADMM	0.232	0.364
4	GLASSO	0.438	0.542

Table 1: Table showing the true positive and false positive rates for all methods. While ISTA and FISTA have the highest true recovery rate, they also have a higher error rate. The subgradient method recovers a very dense graph, while ADMM recovers a very sparse one.

Conclusion: In this project we investigated various optimization methods for the basis pursuit problem and in gaussain graphical models. While all the algorithms have good theoretical properties, it turns out that some of them do work better in some cases than other. ADMM worked better than ISTA/FISTA for an ill-conditioned basis-pursuit problem, while ISTA/FISTA had the best performance for Gaussian Graphical Models. Hence the basic conclusion is that, different algorithms tend to work well in different scenarios in practice. In addition, ISTA/FISTA have convergence rates guarantees seemed to hold in general, which is missing in the ADMM framework. In fact, in our simulations, ISTA performed almost as well as FISTA. (All the code was written by me and can be made available upton request.)

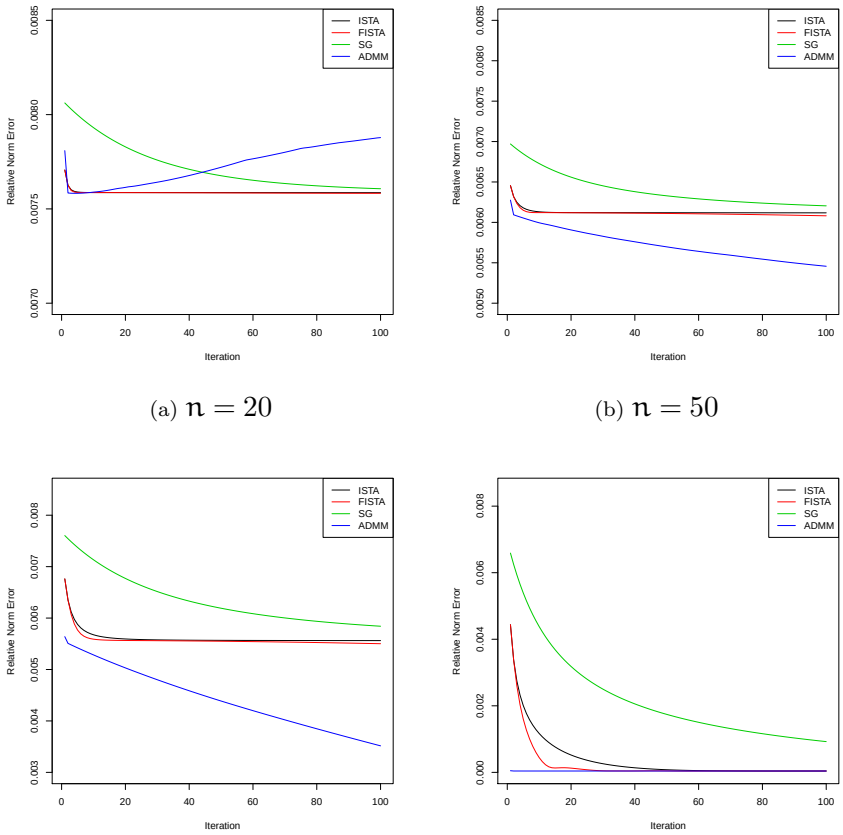


(a) $n = 20$ (b) $n = 50$



(c) $n = 100$ (d) $n = 500$

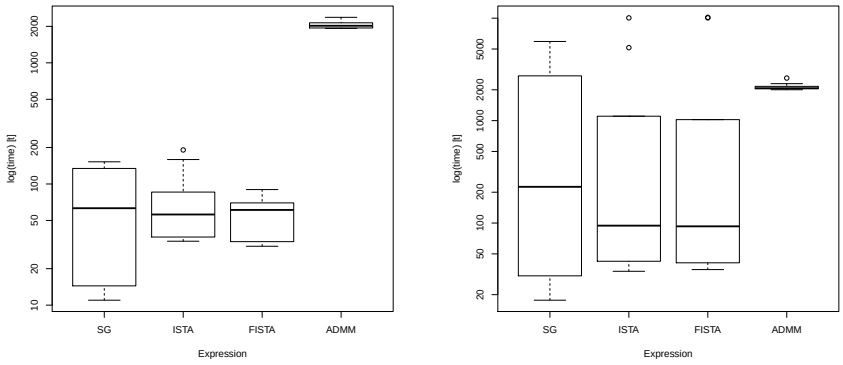
Figure 1: $||\beta^k - \beta^{k-1}||_\infty$ for the well-conditioned X. This checks for convergence of the algorithm. It is clear that according to this measure, ISTA and FISTA perform the best for $n < p$, while ADMM performs the best for the $n > p$ case.



(a) $\mathfrak{n} = 20$ (b) $\mathfrak{n} = 50$

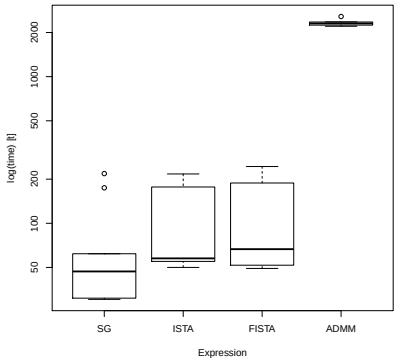
(c) $\mathfrak{n} = 100$ (d) $\mathfrak{n} = 500$

Figure 2: Relative Normed Error from the true solution for the well conditioned X matrix. This checks for convergence to the true solution. At $\mathfrak{n} = 20$, the performance of ADMM is odd because it starts to rise after initially declining. This is not the case for all the other values of $\mathfrak{n}$, where it clearly outperforms the other methods.

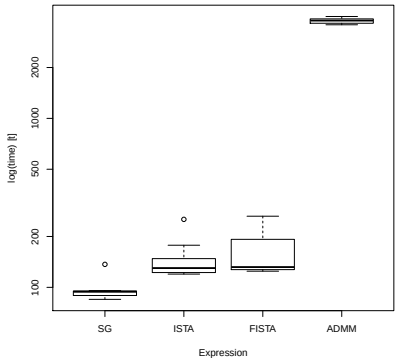


(a) $\mathfrak{n} = 20$

(b) $\mathfrak{n} = 50$

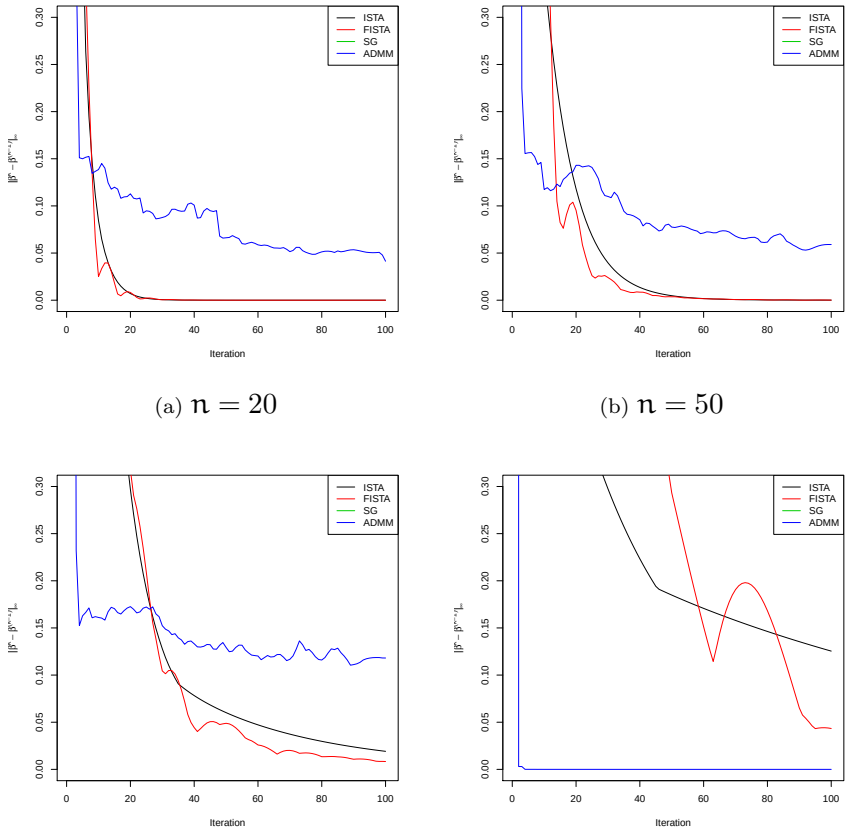


(c) $\mathfrak{n} = 100$



(d) $\mathfrak{n} = 500$

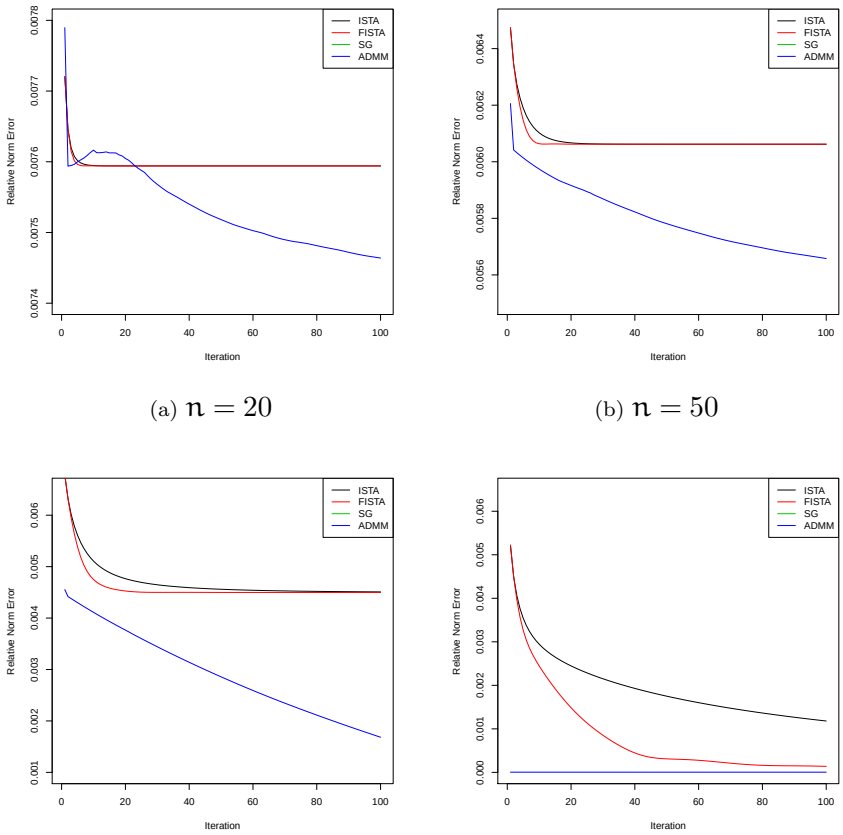
Figure 3: Timing for Basis Pursuit for well-conditioned X. While it is clear that ADMM is the slowesty method, the timing for the other methods is all over the place. Hence it is difficult to find a clear winner in terms of the timing.



(a) $n = 20$ (b) $n = 50$

(c) $n = 100$ (d) $n = 500$

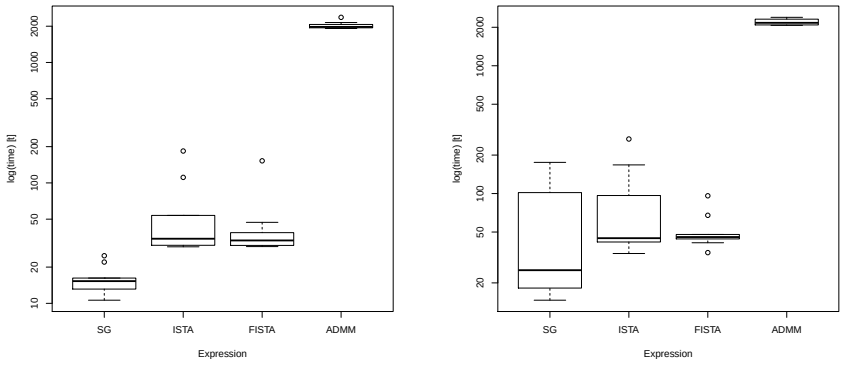
Figure 4: $\left|\left|\beta^k - \beta^{k-1}\right|\right|_\infty$ for the ill-conditioned X. This checks for convergence of the algorithm. The sub-gradient algorithm diverges in this case and hence doesn't appear on the plots. It is clear that according to this measure, ISTA and FISTA performs erratically, but is still the best for $n < p$, while ADMM performs the best for the $n > p$ case.



(a) $\mathfrak{n} = 20$ (b) $\mathfrak{n} = 50$

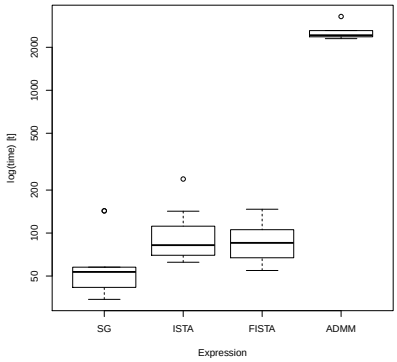
(c) $\mathfrak{n} = 100$ (d) $\mathfrak{n} = 500$

Figure 5: Relative Normed Error from the true solution for the ill-conditioned X matrix. This checks for convergence to the true solution. ADMM clearly performs the best in this case.

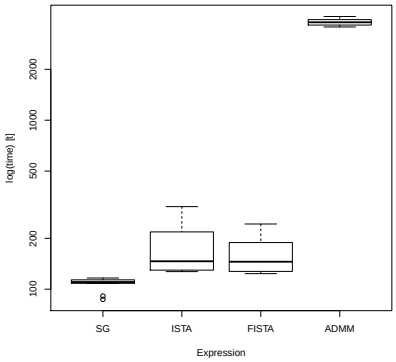


(a) $\mathfrak{n} = 20$

(b) $\mathfrak{n} = 50$



(c) $\mathfrak{n} = 100$



(d) $\mathfrak{n} = 500$

Figure 6: Timing for Basis Pursuit for ill-conditioned X. It is clear that ADMM is the slowesty method with the other methods performing much better. However, in terms of the other measures, their performance was poorer as compared to the ADMM.

References

[1] Beck, A. and Teboulle, M. (2009), "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems, *Siam J. Imaging Sciences* Vol. 2, No. 1, pp. 183-202

[2] Boyd, S. and Parikh, N. and Chu, E. and Peleato, B. (2011), "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, Vol. 3, No. 1, 1-122

[3] Boyd, S. and Vandenberghe, L. (2009), "Convex Optimization,"