Project Proposal

Syed Rahman, Nikou Jah

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

$$\begin{aligned} &\arg\min_{\beta} f(\beta) \\ &= \arg\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \end{aligned} \tag{1}$$

The algorithms we will be looking at are the ISTA, FISTA, ADMM, Split Bregman methods (Syed Rahman) and the Message Passing Algorithm (Nikou Jah). 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.

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

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

For the subgradient, we will just use

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

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 $\|\beta^{new} - \beta^{old}\|_\infty \geq \epsilon$ **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 $\|\beta^{new} - \beta^{old}\|_\infty \geq \epsilon$ **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*

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

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

where $|F(\beta^1) - F(\beta^2)| \leq L\|\beta^1 - \beta^2\|$

Theorem 2 *For diminshing step sizes, the sudgradient method satisfies*

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

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

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

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

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

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

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

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

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 $g(\beta)$:

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

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

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

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.

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

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 $\|\beta^{new} - \beta^{old}\|_\infty \geq \epsilon$ **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

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

Now if

$$F(\beta^{new}) > Q_L(\beta^{old}, \beta^{new})$$

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

$$\iff g(\beta^{new}) > g(\beta^{old}) + < \beta^{new} - \beta^{old}, \nabla g(\beta^{old}) > + \frac{L}{2}\|\beta^{new} - \beta^{old}\|_2^2$$

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 $F(\beta^{new}) \leq Q_{L^{new}}(\beta^{new}, \beta^{old})$ 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 $\|\beta^{new} - \beta^{old}\|_\infty \geq \epsilon$ **do**
  Find smallest integer i such that $F(\beta^{new}) \leq Q_{L^{new}}(\beta^{new}, \beta^{old})$ 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

The following theorem talks about the convergence rates of ISTA:

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

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

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

The following theorem talks about the convergence rates of FISTA:

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

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

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

Algorithm 5 FISTA with fixed step size

Set $t \leftarrow \frac{1}{\lambda_{max}(X^tX)}$
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 $\|\beta^{new} - \beta^{old}\|_\infty \geq \epsilon$ **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

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

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

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

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

Similarly, FISTA would be

1. $\Omega^{new} = S_{\lambda t}(\zeta^{old} + t((\zeta^{old})^{-1} - S))$
2. $\zeta^{old} = \Omega^{new} + \frac{k-1}{k-2}(\Omega^{new} - \Omega^{old})$

In both the above formulations the soft-thresholding is only applied to the off-diagonal elements of the matrices after initializing $\Omega_0 = \text{diag}(S) + \lambda I$

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 $F(\beta^{new}) \leq Q_{L^{new}}(\beta^{new}, \beta^{old})$ 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 $\|\beta^{new} - \beta^{old}\|_\infty \geq \epsilon$ **do**
    $\beta^{old} \leftarrow \beta^{new}$
    Find smallest integer i such that $F(\beta^{new}) \leq Q_{L^{new}}(\beta^{new}, \beta^{old})$ 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

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. Note that we can restate Equation 1 of

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

as to solve this using ADMM we look at the augmented Lagrangian for:

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

which is

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

The ADMM updates in this case are:

1. $\beta^k = \arg\min_\beta L(\beta^{k-1}, \gamma, \eta)$
2. $\gamma^k = \arg\min_\gamma L(\beta, \gamma^{k-1}, \eta)$
3. $\eta^k = \eta^{k-1} + \rho(\beta - \gamma)$

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 $\|\beta^{new} - \beta^{old}\|_\infty \geq \epsilon$ **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

Split Bregman For this part, we will refer to *Bregman Iterative Algorithms for ℓ_1-Minimization with Applications to Compressed Sensing* by Wotao Yin, Stanley Osher, Donald Goldfarb and Jerome Darbon. The basic idea is similar to ADMM. Note that the problem from Equation 1 can be restated as

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

Then the basic updates are:

1. $(\beta^k, u^k) = \arg\min_{\beta,u} \frac{1}{2}\|y - X\beta^{k-1}\|_2^2 + \lambda\|u^{k-1}\|_1 + \frac{\mu}{2}\|\beta^{k-1} - u^{k-1} - b^{k-1}\|_2^2$
2. $b^k = b^{k-1} + (\beta^k - u^k)$

For solving sparse group lasso problems it may be more useful. Suppose $\beta = \{\beta_g\}$ for $g \in G$.

$$\begin{aligned} &\min_\beta f_2(\beta) \\ =&\min_\beta \frac{1}{2}\|y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2 \sum_{g\in G}\|\beta_g\|_2 \end{aligned} \tag{2}$$

This can be restated as

$$\min_{\beta,u,v} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda_1\|u\|_1 + \lambda_2 \sum_{g\in G}\|v_g\|_2 + \frac{\mu_1}{2}\|\beta - u\|_2^2 + \frac{\mu_2}{2}\|\beta - v\|_2^2$$

Then the basic updates are:

1. $(\beta^k, u^k, v^k) = \arg\min_{\beta,u,v} \frac{1}{2}\|y - X\beta^{k-1}\|_2^2 + \lambda_1\|u^{k-1}\|_1 + \lambda_2 \sum_{g\in G}\|v_g^{k-1}\|_2 + \frac{\mu_1}{2}\|\beta^{k-1} - u^{k-1} - b^{k-1}\|_2^2 + \frac{\mu_2}{2}\|\beta^{k-1} - v^{k-1} - c^{k-1}\|_2^2$
2. $b^k = b^{k-1} + (\beta^k - u^k)$
3. $c^k = c^{k-1} + (\beta^k - v^k)$

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

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

which is equivalent to

$$\begin{aligned} &\min_{\Omega,Z} \operatorname{trace}(S\Omega) - \log|\Omega| + \lambda\|Z\|_1 \text{ s.t } \Omega = Z \\ \iff &\min_{\Omega,Z,y} \operatorname{trace}(S\Omega) - \log|\Omega| + \lambda\|Z\|_1 + y^t(\Omega - Z) + \frac{\rho}{2}\|\Omega - Z + U\|_F^2 \end{aligned}$$

Approximate Message passing algorithm Compressed sensing is a framework of techniques which try to estimate high dimensional sparse vectors correctly. Most of these methods are very costly in the sense of computation because they need nonlinear scheme to recover unknown vectors. One class of these schemes used is linear programming(LP) methods to estimate sparse vectors. These methods, unlike the linear methods, are very expensive when you need to recover very huge number of unknown variables with thousands of constraints. The Message Passing Algorithm improves on these LP methods by using belief propagation theory in graphical models.

The Message Passing Algorithm is basically trying to find the answer to the problem of finding $\mu_i(x_i)$ or generally $\mu_S(x_S)$ when $\mu(x_1, x_2, ..., x_n)$ is given. By propagation we can attack this problem in some practical cases. Suppose

$$\mu(x_1, x_2, ..., x_n) = \frac{1}{Z} \prod_{a \in F} \psi_a(x_{S_a})$$

where $S_i \subset \{x_1, ..., x_n\}$ and $S_i \cap S_J \neq \emptyset$ for some i and j. We can define a bipartite graph where $x_1, ..., x_n$ is corresponding to variable nodes and $\psi_a(x_{S_1}), \psi_b(x_{S_2}), ...$ is corresponding to factor nodes and there is an edge between factor node a and variable node i if $x_i \in S_a$. If we define $\partial a = \{i : x_i \in S_a\}$ and $\partial i = \{b : i \in S_b\}$ we are interested to find

$$\mu_{a \to j}(x_j) = \sum_{\{x_j : j \in \partial a \backslash j\}} \psi_a(x_{\partial a}) \prod_{\{l \in \partial a \backslash j\}} \mu'_{l \to a}(x_l)$$

and

$$\mu'_{j \to a}(x_j) = \prod_{\{b \in \partial j \backslash a\}} \mu_{b \to j}(x_j).$$

If we consider the iterative algorithm

$$v^{t+1}_{a \to j}(x_j) = \sum_{\{x_j : j \in \partial a \backslash j\}} \psi_a(x_{\partial a}) \prod_{\{l \in \partial a \backslash j\}} v'^{t}_{l \to a}(x_l)$$

and

$$v'^{t+1}_{j \to a}(x_j) = \prod_{\{b \in \partial j \backslash a\}} v^t_{b \to j}(x_j)$$

under some circumstances

$$v^t_{b \to j}(x_j) \to \mu_{b \to j}(x_j)$$

and

$$v'^{t}_{b\to j}(x_j) \to \mu'_{j\to a}(x_j)$$

when $t \to \infty$. If μ is pdf on $\mathbb{R}^n$ we can extend the algorithm to

$$v^{t+1}_{a\to j}(x_j) = \int_{\{x_j : j\in\partial a\setminus j\}} \psi_a(x_{\partial a}) \prod_{\{l\in\partial a\setminus j\}} v'^{t}_{l\to a}(x_l)$$

and

$$v'^{t+1}_{j\to a}(x_j) = \prod_{\{b\in\partial j\setminus a\}} v^{t}_{b\to j}(x_j).$$

Recall that the lasso problem is to minimize

$$\frac{1}{2} \parallel y - Ax \parallel_2^2 + \lambda \parallel x \parallel_1 .$$

If we define

$$\mu(dx) = \frac{1}{Z} e^{\frac{-\beta}{2}\|y-Ax\|_2^2 - \beta\lambda\|x\|_1} dx$$

where $\beta > 0$, as $\beta \to \infty$, μ concentrates around the solution of lasso x'^{λ}.Therefore by following the steps of message passing algorithm we can find the approximate solution of the lasso problem. This algorithm is called Approximate Message algorithm(AMS). As mentioned earlier, we will be reviewing this algorithm steps and analyzing its weaknesses and strengths.