Project Proposal

Syed Rahman, Nikou Sefat

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 Sefat). 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. If time allows, we will explore applications to gaussian graphical models.

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

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

where $[S_{\lambda t}(x)]_i = \text{sign}(x_i) * \max\{|x_i| - \lambda t, 0\}$. 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:

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

and the FISTA update is as follows:

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

Finally, we will discuss convergence properties for the back-tracking line search.

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

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. $c^k = c^{k-1} + (\beta^k - v^k)$

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

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

and

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

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.