10-725/36-725: Convex Optimization **Fall 2019**

Lecture 21: November 6 ADMM

Lecturer: Lecturer: Ryan Tibshirani *Scribes: Scribes: Motolani Olarinre, Derun Gu, Jingxiao Liu*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

This lecture's notes illustrate some uses of various LaTeX macros. Take a look at this and imitate.

21.1 Recap: Dual Decomposition

Consider problem:

$$\min_x f(x) \quad \text{subject to} \quad Ax = b$$

where f is strictly convex and closed (i.e. its conjugate is differentiable). if $f(x)$ can be decomposed as $f(x) = \sum_{i=1}^{B} f_i(x_i)$, then we can apply dual ascent and obtain the updates for the primal variable as:

$$x^+ = \arg\min_x \sum_{i=1}^{B} f_i(x_i) + u^T Ax$$

$$\iff x_i^+ = \arg\min_{x_i} f_i(x_i) + u^T A_i x_i \quad i = 1, ..., B$$

This minimizer x^+ is the gradient of the dual of the original problem at point u (where is the dual variable). Augmented Lagrangian method (method of multipliers) imposes strong convexity on the primal by adding the term $\frac{\rho}{2} \|Ax - b\|_2^2$. This improves the convergence guarantees of the dual ascent method.

The Alternating Direction Method of Multipliers (ADMM), can be applied to problems of the form:

$$\min_{x,z} f(x) + g(z) \quad \text{subject to} \quad Ax + Bz = c$$

The augmented lagangian looks like:

$$L_p(x, z, u) = f(x) + g(z) + u^T(Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|_2^2$$

At each step or ADMM, we alternately minimize the augmented lagrangian wrt primal variables x and z, then update dual variable u with the minimizers. Scaled ADMM defines a new dual variable $w = \frac{u}{\rho}$ and then uses this in the augmented lagrangian, as well as for dual variable updates.

21.2 ADMM: Connection to proximal operators

Consider:

$$\min_x f(x) + g(x) \iff \min_{x,z} f(x) + g(z) \quad \text{subject to} \quad x = z$$

The scaled form of the gradient update with primal variable x becomes:

$$x^+ = \arg\min_x f(x) + \frac{\rho}{2}\|x - z + w\|_2^2$$

$$= \arg\min_x \frac{1}{2(\frac{1}{\rho})}\|z - w - x\|_2^2 + f(x) = prox_{f,\frac{1}{\rho}}(z - w)$$

Where $prox_{f,\frac{1}{\rho}}$ is the proximal operator for f at parameter $\frac{1}{\rho}$. The z gradient update is similar. Therefore we get the following ADMM update steps:

$$x^{(k)} = prox_{f,\frac{1}{\rho}}(z^{(k-1)} - w^{(k-1)})$$

$$z^{(k)} = prox_{g,\frac{1}{\rho}}(x^{(k)} + w^{(k-1)})$$

$$w^{(k)} = w^{(k-1)} + x^{(k)} - z^{(k)}$$

This algorithm is called Douglas-Rachford. In the original minimization problem, if $f(x)$ is smooth, we can apply proximal gradient method. If however $f(x)$ is not smooth, we can generalize proximal gradient method with Douglas-Rachford algorithm.

21.3 ADMM Example

21.3.1 Lasso Regression

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

We can reparameterize this as:

$$\min_{\beta,\alpha} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\alpha\|_1 \quad \text{subject to} \quad \beta - \alpha = 0$$

We can compute the β that minimizes the augmented lagrangian as:

$$\beta^+ = \arg\min_\beta \frac{1}{2}\|y - X\beta\|_2^2 + \frac{\rho}{2}\|\beta - \alpha + w|_2^2$$

$$= (X^TX + \rho I)^{-1}(X^Ty + \rho(\alpha - w))$$

This can be viewed as ridge regression with an offset. The α updates are:

$$\alpha^+ = \arg\min_\alpha \lambda\|\alpha\|_1 + \frac{\rho}{2}\|\beta - \alpha + w|_2^2$$

$$= \arg\min_\alpha \frac{\lambda}{\rho}\|\alpha\|_1 + \frac{1}{2}\|\beta + w - \alpha|_2^2$$

$$= S_{\frac{\lambda}{\rho}}(\beta + w) \leftarrow \text{ (soft-thresholding operator)}$$

The dual variable updates are: $w^+ = w + \beta - \alpha$
In this case, ADMM in essence does ridge regression and then thresholds. Does this in sequence makes in equivalent to Lasso.

21.3.2 Practicalities

Like first order methods, ADMM can obtain a relatively accurate solution in a handful of iterations, but requires a large number of iterations for a high accuracy solution.
Choice of ρ: ρ balances the contributions of residual vs objective in the criterion, and influences the speed of convergence for ADMM. If ρ is too large, too much emphasis is placed on the residual, and if ρ is too small, too much emphasis is placed on the objective.
Steven Boyd et al (2010) developed a heuristic for picking ρ at each step of ADMM. At each step, compute the primal and dual residuals as follows:

$$\text{primal residual: } Ax + Bz - c$$

$$\text{dual residual: } \begin{bmatrix} \nabla f(x) & A^T u \\ \nabla g(z) & B^T u \end{bmatrix}$$

Compare the residuals at each step. Increase ρ if primal residual is very large compared to dual residual and vice versa. This works well in practice but gives no convergence guarantees.

21.3.3 Group lasso regression

The group lasso regression has the form as below.

Given $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times p}$, we want to do the minimization:

$$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda \sum_{g=1}^{G} c_g \|\beta_g\|_2.$$

Rewrite the question:

$$\min_{\beta,\alpha} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda \sum_{g=1}^{G} c_g \|\alpha_g\|_2,$$
$$s.t. \quad \beta - \alpha_g = 0 \ \ \forall\ g \in [G].$$

Now optimizing β, we have

$$\beta^{(k)} = \arg\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \frac{\rho}{2}\|\beta - (\alpha^{(k-1)} - w^{(k-1)})\|_2^2.$$

Take derivative on the right-hand-side and set it to zero, we have

$$X^T(X\beta - y) + \rho(\beta - (\alpha^{(k-1)} - w^{(k-1)})) = 0.$$

Therefore, we have

$$\beta^{(k)} = (X^T X + \rho I)^{-1}(X^T y + \rho(\alpha^{(k-1)} - w^{(k-1)})).$$

Notice that $X^T X + \rho I$ is always invertible, and we can pre-calculate it to save time.

Now we consider α. The ADMM update rule for α is

$$\alpha^k = \arg\min_{\alpha} \lambda \sum_{g=1}^{G} c_g \|\alpha_g\|_2 + \frac{\rho}{2}\|\beta^{(k)} - \alpha + w\|_2^2.$$

Take derivative w.r.t. α_g and set it to zero, we have

$$\lambda c_g \frac{\alpha_g}{\|\alpha_g\|_2} - \rho(\beta_g^{(k)} - \alpha_g + w_g^{(k-1)}) = 0$$

This gives

$$\alpha_g^{(k)} = R_{c_g\lambda/\rho}(\beta_g^k + w_g^{(k-1)}),$$

where

$$R_t(x) = (1 - \frac{t}{\|x\|_2})x.$$

Finally, we have

$$w^{(k)} = w^{(k-1)} + \beta^{(k)} - \alpha^{(k)}.$$

Notice that if groups can overlap with each other, the above ADMM algorithm can be slightly modified to apply to this new problem, while it may be very difficult for other optimization methods to solve the new overlapped lasso problem.

21.3.4 Sparse subspace estimation

The sparse subspace estimation has the form

$$\max_Y \quad tr(SY) - \lambda\|Y\|_1 \quad s.t. \quad Y \in \mathcal{F}_k$$

where $\mathcal{F}_k$ is the Fantope of order k, namely

$$\mathcal{F}_k = \{Y \in \mathbb{S}^p | 0 \prec Y \prec I, tr(Y) = k\}.$$

We can view this problem as a penalized PCA form due to the following intuition:

$$\begin{aligned}
&\min_R \|X - R\|_F^2 \quad s.t. \quad rank(R) = k \quad \text{(The original PCA problem)} \\
\iff &\min_P \|X - XP\|_F^2 \quad s.t. \quad P \text{ is a projection matrix and } rank(P) = k \\
\iff &\max_P < X^T X, P > \quad s.t. \quad P \in C_k \ (C_k \text{ are projection matrices of rank k}) \\
\iff &\max_P < X^T X, P > \quad s.t. \quad P \in Conv(C_k) \text{ (convex hull of } C_k \text{ , also called } \mathcal{F}_k)
\end{aligned}$$

We can rewrite this problem as

$$\min_{Y,Z} \ -tr(SY) + \mathbb{I}_{\mathcal{F}_k}(Y) + \lambda\|Z\|_1, \quad s.t. \quad Y = Z.$$

The ADMM steps are then given by

$$\begin{aligned}
Y^{(k)} &= P_{\mathcal{F}_k}(Z^{(k-1)} - W^{(k-1)} + \frac{1}{\rho}S) \\
Z^{(k)} &= S_{\lambda/\rho}(Y^{(k)} + W^{(k-1)}) \\
W^{(k)} &= W^{(k-1)} + Y^{(k)} - Z^{(k)}
\end{aligned}$$

Here $P_{\mathcal{F}_k}$ is Fantope projection operator.

If $A = U\Sigma U^T, \Sigma = (\sigma_1, \ldots, \sigma_p)$, then

$$P_{\mathcal{F}_k}(A) = U\Sigma_\theta U^T, \Sigma_\theta = (\sigma_1(\theta), \ldots, \sigma_p(\theta))$$

where $\sigma_i(\theta) = \min\{\max\{\sigma_i - \theta, 0\}, 1\}$ and $\sum_{i=1}^p \sigma_i(\theta) = k$.

21.3.5 Sparse + low rank decomposition

The sparse + low rank decomposition problem has the following form:

$$\min_{L,S} \|L\|_{tr} + \lambda\|S\|_1 \quad s.t. \quad L + S = M.$$

This problem can be transformed to SDP problem and solve by interior point method, but requires significant amount of efforts.

On the contrary, ADMM provides an easier approach that has the following steps:

$$L^{(k)} = S^{tr}_{1/\rho}(M - S^{(k-1)} + W^{(k-1)})$$
$$S^{(k)} = S^{\ell_1}_{\lambda/\rho}(M - L^{(k)} + W^{(k-1)})$$
$$W^{(k)} = W^{(k-1)} + M - L^{(k)} - S^{(k)}$$

Here S^{tr} represents matrix soft-thresholding and S^{ℓ_1} represents elementwise soft-thresholding.

This problem was proposed by Candes et. al. in 2009 where he gave an example of its application in video surveillance as shown in 21.1.

21.4 Consensus ADMM

Consider a general problem

$$\min_x \sum_{i=1}^B f_i(x).$$

The consensus ADMM approach begins by reparametrizing the above problem to the following form:

$$\min_{x_1,\ldots,x_B,x} \sum_{i=1}^B f_i(x_i) \quad s.t. \quad x_i = x \quad \forall\ i \in [B].$$

By such transformation, the updates of x_i at each ADMM step are independent and therefore can be run in parallel.

The detailed ADMM steps:

$$x_i^{(k)} = \arg\min_{x_i} f_i(x_i) + \frac{\rho}{2}\|x_i - x^{(k-1)} + w_i^{(k-1)}\|_2^2 \quad i = 1, \ldots, B$$
$$x^{(k)} = \frac{1}{B}\sum_{i=1}^B (x_i^{(k)} + w_i^{(k-1)})$$
$$w_i^{(k)} = w_i^{(k-1)} + x_i^{(k-1)} - x^{(k)} \quad i = 1, \ldots, B$$

(a) Original frames (b) Low-rank $\hat{L}$ (c) Sparse $\hat{S}$

Figure 21.1: Example decomposition of video surveillance data. The low-rank matrix corresponds to the background while the sparse matrix corresponds to the moving objects.

Let $\bar{x} = \frac{1}{B}\sum_{i=1}^{B} x_i$.

Notice that for any iteration $k \geq 1$,

$$\bar{w}^{(k)} = \frac{1}{B}\sum_{i=1}^{B} w_i^{(k)} = \frac{1}{B}\sum_{i=1}^{B}(x_i^{(k)} + w_i^{(k-1)}) - x^{(k)} = x^{(k)} - x^{(k)} = 0.$$

Therefore, for any iteration $k \geq 2$, the ADMM steps can be simplified as:

$$x_i^{(k)} = \arg\min_{x_i} f_i(x_i) + \frac{\rho}{2}\|x_i - \bar{x}^{(k-1)} + w_i^{(k-1)}\|_2^2, \quad i = 1, \ldots, B$$
$$w_i^{(k)} = w_i^{(k-1)} + x_i^{(k)} - \bar{x}^{(k)}, \quad i = 1, \ldots, B$$

Each of those two steps can be run in parallel to accelerate computation.

In general, the consensus ADMM algorithm solves the following problem:

$$\min_{x} \quad \sum_{i=1}^{B} f_i(a_i^T x + b_i) + g(x).$$

The reparameterized problem has form:

$$\min_{x_1,\ldots,x_B,x} \sum_{i=1}^{B} f_i(a_i^T x_i + b_i) + g(x), \quad s.t. \quad x_i = x, \quad \forall\, i \in [B].$$

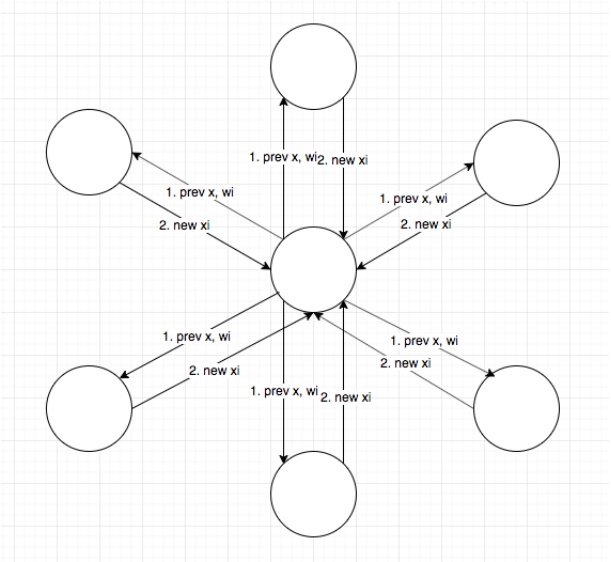

Figure 21.2: A graph illustration of consensus ADMM algorithm

The ADMM updates are:

$$
\begin{aligned}
x_i^{(k)} &= \arg\min_{x_i} f_i(a_i^T x_i + b_i) + \frac{\rho}{2}\|x_i - x^{(k-1)} + w_i^{(k-1)}\|_2^2, \quad i = 1, \ldots, B \\
x^{(k)} &= \arg\min_{x} \frac{B\rho}{2}\|x - \bar{x}^{(k)} - \bar{w}^{(k-1)}\|_2^2 + g(x) \\
w_i^{(k)} &= w_i^{(k-1)} + x_i^{(k)} - x^{(k)}, \quad i = 1, \ldots, B
\end{aligned}
$$

Notice that now we cannot simplify the updates as before since $w^{(k)} \neq 0$ in general.

A graph explanation of this general algorithm is shown in figure 21.2.

During the first step, the server first sends $x^{(k-1)}$ and $w_i^{(k-1)}$ to every nodes, and each nodes in charge of computing an $x_i^{(k)}$. Then the server aggregates all responses of $x_i^{(k)}$ from all nodes.

During the second and third steps, the server computes $x^{(k)}$ and $w_i^{(k)}$.

Since we distribute the most computational heavy step, i.e., the first step, the consensus ADMM algorithm is very efficient in general.

21.5 Spatial decompositions for ADMM

ADMM can exhibit much faster convergence than usual, when we parametrize subproblems in a "spetial way". ADMM updates relate closely to block coordinate descent, in which we optimize a criterion in an alternating fashion across blocks of variables. With this in mind, get fastest convergence when minimizing over blocks of variables leads to updates in nearly orthogonal directions. This suggests we should design ADMM form so that primal updates de-correlate as best as possible. This work has been done in e.g., Ramdas and Tibshirani (2014), Wytock et al. (2014), Barbero and Sra (2014).

21.5.1 Example: 2d fused lasso

Given an image $Y \in \mathcal{R}^{d \times d}$, equivalently written as $y \in \mathcal{R}^n$, the 2d fused lasso or 2d total variation denoising problem is

$$\min_{\Theta} \frac{1}{2}\|Y - \Theta\|_F^2 + \lambda \sum_{i,j} (|\Theta_{i,j} - \Theta_{i+1,j}| + |\Theta_{i,j} - \Theta_{i,j+1}|)$$

$$\iff \min_{\theta} \frac{1}{2}\|y - \theta\|_2^2 + \lambda \|D\theta\|_1$$

where Θ represents the parameter matrix for each pixel in the image. $D \in \Updownarrow \times \setminus$ is a 2d difference operator giving the appropriate differences (across horizontally and vertically adjacent positions). Figure 21.3 shows the penalty terms. Neighboring pixels should be assigned to the same value, in order to avoid fusions in both the horizontal and vertical directions.

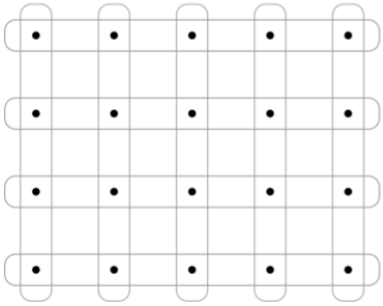

Figure 21.3: .The penalty terms in 2d fused lasso

Rewrite the problem:

$$\min_{\theta z} \frac{1}{2}\|y - \theta\|_2^2 + \lambda \|z\|_1 \quad \text{subject to } \theta = Dz$$

This lead to the following ADMM steps:

$$\begin{aligned} \theta^{(k)} &= (I + \rho D^T D)^{-1}(y + \rho D^T (z^{(k-1)} + w^{(k-1)})) \\ z^{(k)} &= S_{\lambda/\rho}(D\theta^{(k)} - w^{(k-1)}) \\ w^{(k)} &= w^{(k-1)} + z^{(k-1)} - D\theta^{(k)} \end{aligned} \tag{21.1}$$

The θ update solves linear system in $I + \rho L$, with $L = D^T D$ the graph Laplacian matrix of the 2d grid, so this can be done efficiently, in roughly $O(n)$ operations. The z update applies soft thresholding operator S_t. Hence one entire ADMM cycle uses roughly $O(n)$ operations.

We can also rewite the problem as

$$\min_{H,V} \frac{1}{2}\|Y - H\|_F^2 + \lambda \sum_{i,j} (|H_{i,j} - H_{i+1,j} + |V_{i,j} - V_{i,j+1}|) \quad \text{subject to } H = V$$

This leads to ADMM steps:

$$\begin{aligned} H_{\cdot,j}^{(k)} &= FL_{\lambda/(1+\rho)}^{1d}\Big(\frac{Y+\rho(V_{\cdot,j}^{(k-1)}-W_{\cdot,j}^{(k-1)})}{1+\rho}\Big), \quad j=1,\cdots,d \\ V_{i,\cdot}^{(k)} &= FL_{\lambda/\rho}^{1d}(H_{i,\cdot}^{(k)}+W_{i,\cdot}^{(k-1)}), \quad i=1,\cdots,d \\ W^{(k)} &= W^{(k-1)}+H^{(k)}-V^{(k)} \end{aligned} \tag{21.2}$$

Both H, V updates solve sequence of 1d fused lassos, where we write $FL_\tau^{1d}(a) = \arg\min_x \frac{1}{2}\|a-x\|_2^2 + \tau\sum_{i=1}^{d-1}|x_i - x_{i+1}|$. Each 1d fused lasso solution can be computed exactly in $O(d)$ operations with specialized algorithms. Hence, one entire ADMM cycle again uses $O(n)$ operations.

21.6 References

A. Barbero and S. Sra (2014), "Modular proximal optimization for multidimensional total-variation regularization"

A. Ramdas and R. Tibshirani (2014), "Fast and flexible ADMM algorithms for trend filtering"

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine learning, 3(1), 1-122.

Candès, E. J., Li, X., Ma, Y., Wright, J. (2011). Robust principal component analysis?. Journal of the ACM (JACM), 58(3), 11.

M. Wytock and S. Sra. and Z. Kolter (2014), "Fast Newton methods for the group fused lasso"