ADMM Review

Yue Zhang

This review is based on but not limited to the paper:
'Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers' by Stephen Boyd et al.
(* Questions are appreciated *)

June 4, 2015

Outline

Background

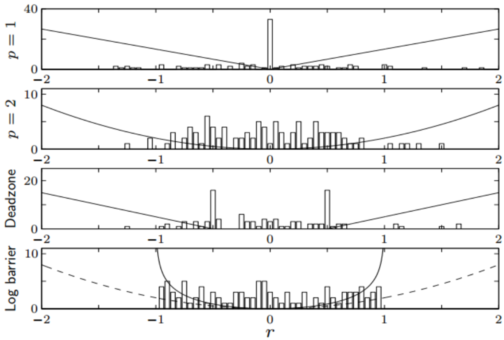
ADMM

L1 minimization

- Why do we do ℓ_1 minimization? Sparsity? Why?
- Consider minimize $\sum \phi(r_i)$, subject to $r = Ax - b$, where $A \in R^{m\times n}$, $b \in R^m$. (From Boyd's class EE364a, Lec6)

example ($m = 100$, $n = 30$): histogram of residuals for penalties

$$\phi(u) = |u|, \quad \phi(u) = u^2, \quad \phi(u) = \max\{0, |u| - a\}, \quad \phi(u) = -\log(1-u^2)$$



Side story

- Can we combine their advantages? Huber penalty function, (mark here, not quite developed yet)
- Easy check with CVX, a Matlab toolbox.

```
A = rand(100,30);
b = rand(100);

cvx_begin
variables r(100) x(30)
minimize( norm(r,1) )
A*x-b==r
cvx_end

hist(r)
```

Outline

Background

ADMM

Precursors

(random selected concepts (should know) before moving forward)

- Equality constrained convex problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b, \end{array}$$

Lagrangian:

$$L(x, y) = f(x) + y^T(Ax - b)$$

Dual function (offers lower bound)

$$g(y) = \inf_x L(x, y) = -f^*(-A^T y) - b^T y$$

$f^*(y)$ is the conjugate function of $f(x)$, defined as $\sup_x (y^T x - f(x))$. Recall Dr. Guo's lecture notes, the image of conjugate function, self-check with CO ex3.36.

Precursors

- If $f(x)$ twice differentiable, strong duality holds (easy check with KKT conditions), of course iff there exists feasible solutions.
- Even strong duality doesn't hold for some cases, it still helps if we know the gap. *e.g.*, interior-point methods, with which we can solve inequality constrained convex problems by building up a barrier near the boundary. More details see Boyd's EE364a Lec12.
- As the dual function offers a lower bound, we'll want to maximize it. Method: gradient ascent.
- $g(y) = \inf_x L(x, y)$ indicates $\nabla g = Ax^* - b$.
- side story, if g is not differentiable, take subgradient. More details see Boyd's EE364b Lec1.

A Primal-Dual algorithm

- A first look framework

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b, \end{array}$$

- Iterative:

$$\begin{aligned} x^{k+1} &:= \arg\min_x L(x, y^k) \\ y^{k+1} &:= y^k + \alpha^k (Ax^{k+1} - b) \end{aligned}$$

- However, the x step doesn't necessarily return a feasible x, *e.g.*, what if $L(x, y)$ is affine in x?

- Augmented Lagrangian:

$$L_\rho(x, y) = f(x) + y^T(Ax - b) + (\rho/2)\|Ax - b\|_2^2.$$

- However, there is a cost: $L(x, y)$ can be seperable if both $f(x)$ and A are seperable, *i.e.* $f(x) = \sum_{i=1}^N f_i(x_i)$, $Ax = \sum_{i=1}^N A_i x_i$ is block diagonal. However, the augmented $L_\rho(x, y)$ won't. (I think this is what Julia means in Ben's defense.)

ADMM

- WLOG, we form the problem:

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array}$$

- Augmented Lagrangian:

$$L_\rho(x, y) = f(x) + g(z) + y^T(Ax + Bz - c) + (\rho/2)\|Ax + Bz - c\|_2^2.$$

- Iterative:

$$\begin{aligned} x^{k+1} &:= \arg\min_x L_\rho(x, z^k, y^k) \\ z^{k+1} &:= \arg\min_z L_\rho(x^{k+1}, z, y^k) \\ y^{k+1} &:= y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \end{aligned}$$

- Note that (z^{k+1}, y^{k+1}) is a function of (z^k, y^k). x^{k+1} is just an intermediate step.
  BTW, as ρ is fixed, its value doesn't make quite a difference (as long as it's reasonable). Boyd chooses ρ to be 1 while Jing chooses 0.01, tested, almost the same performance (iteration numbers and accuracy). Of course this may change with scale of problems.

Convergence

- Under reasonable assumptions: problem solvable, (f,g are nice, L_0 has a saddle point), ADMM iterates satisfy:
  - Residual convergence. $r^k = Ax^k + B^k - c$ converges. (approach feasibility)
  - Objective convergence. $f(x^k) + g(z^k)$ converges to optimal value.
  - Dual variable convergence. y^k converges.

- However, x^k and z^k doesn't necessarily converge. We'll see the reason from the stop criterion.

Stopping Criteria

- Feasibility of primal and dual variables

$$\begin{array}{ll} \text{primal} & Ax^* + Bz^* - c = 0 \\ \text{dual} & \partial f(x^*) + A^T y^* = 0 \\ \text{dual} & \partial g(z^*) + B^T y^* = 0 \end{array}$$

- This can be derived similarly as we did for dual gradient ascent.

- What's interesting is the following, since x^{k+1} minimizes augmented $L_\rho(x, z^k, y^k)$, we have

$$\partial f(x^{k+1}) + A^T y^{k+1} + \rho A^T B(z^k - z^{k+1}) = 0$$

  Which says when dual is feasible, z falls into $A^T B$'s null space, therefore not necessary to converge.

Faster

- Varying penalty parameter. Change ρ to ρ^k,

$$\rho^{k+1} := \begin{cases} \tau^{incr}\rho^k & : \text{if } \|r^k\|_2 > \mu\|s^k\|_2 \\ \rho^k\tau^{decr} & : \text{if } \|s^k\|_2 > \mu\|r^k\|_2 \\ \rho^k & : otherwise, \end{cases}$$

  Where $\mu > 1$, $\tau^{incr} > 1$ and $\tau^{decr} > 1$. Typical choice is $\mu = 10$, $\tau^{incr} = \tau^{decr} = 2$.

- This is due to the different role of ρ played in primal and dual problems. In short word, it cannot be too big or too small.

Extension

- Change augment terms (I have no idea why this is an improvement, mimic conjugate gradient?). Change normal augment term $\|r\|_2^2$ to $r^T P r$, where P is symmetric p.d.

- Over-relaxation. In the $z-$ and $y-$updates, the quantity Ax^{k+1} can be replaced with

$$\alpha^k Ax^{k+1} - (1 - \alpha^k)(Bz^k - c)$$

- Inexact minimization. ADMM converges even $x-$ and $z-$minimization updates don't carry out exactly. That is, when you use iteration methods to minimize the $x-$ and $z-$ subproblems, you can terminate early (if proper). Actually, this is what some people do in large scale problems. See video talk. `http://videolectures.net/nipsworkshops2011_boyd_multipliers/?q=boyd`. It starts at 01:00:40 if you're on hurry.

Related Algorithms

- This is something I want to point out but I don't know anything about them. This is interesting because we may go to its equivalent / or related algorithms, investigate what problems people are dealing with using those algorithms and therefore have a broad idea where ADMM can further apply.
  - operator splitting methods (Douglas, Peaceman, Rachford, Lions, Mercier, . . . 1950s, 1979)
  - proximal point algorithm (Rockafellar 1976)
  - Dykstras alternating projections algorithm (1983)
  - Spingarns method of partial inverses (1985)
  - Rockafellar-Wets progressive hedging (1991)
  - proximal methods (Rockafellar, many others, 1976present)
  - Bregman iterative methods (2008present)
  - most of these are special cases of the proximal point algorithm

Related Problems

- This is something I definitely should point out and I know something(a little) about. (I will write more about these problems in the future)
  - Basis pursuit
  - Lasso (Least absolute shrinkage and selection operator) They just want to make it lasso...
  - Support vector machine (in a sparse view).
  - Sparse Inverse Covariance Selection
  - TGV_α^2 and Dr. Guo's paper, including possible parallel scheme.
  - Sparse modeling, especially sparse coding in deep learning. Some interesting videos may be helpful from `https://www.youtube.com/playlist?list=PLZ9qNFMHZ-A79y1StvUUqgyL-O0fZh2rs`. This is an online course offered by Guillermo Sapiro. Specifically, lectures about sparse modeling and compressive sensing.