Optimization in Big Data Research (III) Alternating Direction of Method of Multipliers - ADMM

Jianyong Sun
School of Mathematics and Statistics
Xi'an Jiaotong University

Xi'an, 2018

Outline

Table of Contents

- convex equality constrained optimization problem

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

- f is separable

$$f(x) = f_1(x_1) + \cdots + f_N(x_N), x = (x_1, \cdots, x_N)$$

- N large

Goals: robust methods for

- arbitrary-scale optimization
  - big data
  - dynamic optimization on large-scale network
- decentralized optimization
  - parallel computing, by passing relatively small messages.

Table of Contents

- convex equality constrained optimization problem

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

- Lagrangian: $L(x, y) = f(x) + y^T(Ax - b)$
- dual function: $g(y) = \inf_x L(x, y)$
- dual problem:

$$\text{maximize } g(y)$$

- recover:

$$x^* = \arg\min_x L(x, y^*)$$

Dual descent

- gradient method for dual problem: $y_{k+1} = y_k + \alpha_k \nabla g(y_k)$
- $\nabla g(y_k) = A\tilde{x} - b$, where $\tilde{x} = \arg\min_x L(x, y_k)$
- dual ascent method is

$$\begin{aligned} x_{k+1} &:= \arg\min_x L(x, y_k) \rightarrow \text{x-minimization} \\ y_{k+1} &:= y_k + \alpha_k (Ax_{k+1} - b) \rightarrow \text{dual update} \end{aligned}$$

- works, but with lots of strong assumptions
  - f be convex, finite and have compact lower level sets.

Dual Decomposition

- if f is separable, then L is separable

$$L(x,y) = L_1(x_1,y) + \cdots + L_N(x_N,y) - y^\mathsf{T} b$$

where $L_i(x_i,y) = f_i(x_i) + y^\mathsf{T} A_i x_i$ and $A = [A_1, A_2, \cdots, A_N]$

- x-minimization in dual ascent splits into N separate minimizations

$$x_{i,k+1} := \arg\min_{x_i} L_i(x_i, y_k)$$

which can be done in parallel.

Dual Decomposition

- dual decomposition

$$
\begin{aligned}
x_i^{k+1} &:= \arg\min_{x_i} L_i(x_i, y^k), i = 1, 2, \cdots, N \\
y^{k+1} &:= y^k + \alpha_k \left(\sum_{i=1}^{N} A_i x_i^{k+1} - b \right)
\end{aligned}
$$

- update x_i in parallel, gather $A_i x_i^{k+1}$; scatter y^k (limited communication among parallel processes)
- To solve a large problem by dual decomposition
  - by iteratively solving the x-minimization subproblems (in parallel)
  - dual variable update provides coordination
- works, but with lots of assumptions; often slow.

Table of Contents

Method of Multipliers

- a method to make dual ascent robust
- based on **augmented Lagrangian** (Hestense, Powell 1969), given $\rho > 0$

$$L_\rho(x, y) = f(x) + y^\mathsf{T}(Ax - b) + \frac{\rho}{2}\|Ax - b\|_2^2$$

- method of multipliers can be formalized as

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

compared to dual decomposition

- converges under much more relaxed conditions (f can be nondifferentiable, can take on value $+\infty$, etc.), but
- quadratic penalty destroys splitting of the x-update, so decomposition is not attainable, thus no good for large scale optimization

Table of Contents

Alternating Direction Method of Multipliers (ADMM)

- ADMM problem form (assume f, g are convex)

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

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

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

- minimize over x and z jointly, ADMM reduces to method of multipliers
- decomposition becomes available on x-minimization and z-minimization
- optimality conditions for differentiable f, g are satisfied by ADMM
  - primal feasibility: $Ax + Bz - c = 0$
  - dual feasibility: $\nabla f(x) + A^{\mathsf{T}} y = 0$, $\nabla g(z) + B^{\mathsf{T}} z = 0$

ADMM with scaled dual variables

- combine linear and quadratic terms in augmented Lagrangian

$$
\begin{aligned}
L_\rho(x,z,y) &= f(x)+g(z)+y^\mathsf{T}(Ax+Bz-c)+\frac{\rho}{2}\|Ax+Bz-c\|_2^2 \\
&= f(x)+g(z)+\frac{\rho}{2}\|Ax+Bz-c+u\|_2^2 + const.
\end{aligned}
$$

with $u=(1/\rho)y$. This holds because (let $r=Ax+Bz-c$)

$$
y^\mathsf{T}r+\frac{\rho}{2}\|r\|_2^2=\frac{\rho}{2}\|r+\frac{1}{\rho}y\|_2^2-\frac{\rho}{2}\|y\|_2^2=\frac{\rho}{2}\|r+u\|_2^2-\frac{\rho}{2}\|u\|_2^2
$$

- ADMM (scaled dual form)

$$
\begin{aligned}
x^{k+1} &:= \operatorname{arg\,min}_x\big(f(x)+(\rho/2)\|Ax+Bz^k-c+u^k\|_2^2\big) \\
z^{k+1} &:= \operatorname{arg\,min}_z\big(g(z)+(\rho/2)\|Ax^{k+1}+Bz-c+u^k\|_2^2\big) \\
u^{k+1} &:= u^k+(Ax^{k+1}+Bz^{k+1}-c)
\end{aligned}
$$

Convergence

- assumptions: f, g convex, closed, proper, L_0 has a saddle point
- then ADMM converges

Related Algorithms

- operator splitting methods
- proximal point algorithm
- Dykstra's alternating projections algorithm
- proximal methods
- Bregman iterative methods
- ...

Common Patterns

- x-update step requires minimizing $f(x) + (\rho/2)\|Ax - v\|^2$ (with $v = Bz^k - c + u^k$, which is constant during x-update)
- similar for z-update
- several special cases come up often, can simplify update by exploit structure in these cases

Decomposition

- suppose f is block-separable

$$f(x) = f_1(x_1) + f_2(x_2) + \cdots + f_N(x_N), x = [x_1, \cdots, x_N]$$

- A is block-separable, i.e. $A^{\mathsf{T}}A$ is block-diagnoal
- then x-update splits into N parallel updates of x_i

Table of Contents

Proximal Operator

- consider x-minimization when $A = I$

$$x^+ = \arg\min_x \left(f(x) + \frac{\rho}{2}\|x - v\|_2^2 \right) = \mathbf{prox}_{f,\rho}(v)$$

  where $v = -Bz + c - u$

- some special cases

$$\begin{aligned} f = I_C(\text{indicator fct. of } C) \qquad & x^+ := \Pi_C(v)(\text{projection onto } \; C) \\ f = \lambda\|\cdot\|_1(\ell_1 \text{ norm}) \qquad & x_i^+ := S_{\lambda/\rho}(v_i)(\text{soft thresholding}) \end{aligned}$$

  where C is closed, non-empty and convex, and

$$S_a(v) = (v - a)_+ - (-v - a)_+$$

Quadratic Objective

- $f(x) = 1/2x^\intercal Px + q^\intercal x + r$
- $x^+ := (P + \rho A^\intercal A)^{-1}(\rho A^\intercal v - q)$
- use matrix inversion lemma when computationally advantageous

$$(P + \rho A^\intercal A)^{-1} = P^{-1} - \rho P^{-1}A^\intercal(I + \rho AP^{-1}A^\intercal)^{-1}AP^{-1}$$

- (direct method) cache factorization $P + \rho A^\intercal A$ or $I + \rho AP^{-1}A^\intercal$
- (iterative method) warm start, early stopping, reducing tolerances.

Constrained convex optimization

- consider ADMM for generic problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{C} \end{array}$$

- ADMM form: take g to be indicator of $\mathcal{C}$, i.e. $g(z) = I_C(z)$

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & x - z = 0 \end{array}$$

- algorithm

$$\begin{aligned} x^{k+1} &:= \operatorname{arg\,min}_x \left(f(x) + \frac{\rho}{2} \|x - z^k + u^k\|_2^2 \right) \\ z^{k+1} &:= \Pi_{\mathcal{C}}(x^{k+1} + u^k) \\ u^{k+1} &:= u^k + x^{k+1} - z^{k+1} \end{aligned}$$

Lasso

- lasso problem

$$\text{minimize } \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1$$

- ADMM form

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|z\|_1 \\ \text{subject to} & x - z = 0 \end{array}$$

- algorithm

$$\begin{aligned} x^{k+1} &:= (A^\mathsf{T}A + \rho I)^{-1}(A^\mathsf{T}b + \rho z^k - u^k) \\ z^{k+1} &:= S_{\lambda/\rho}(x^{k+1} + u^k/\rho) \\ u^{k+1} &:= u^k + \rho(x^{k+1} - z^{k+1}) \end{aligned}$$

Sparse inverse covariance selection

- S : empirical covariance of samples from $\mathcal{N}(0, \Sigma)$, with Σ^{-1} sparse (i.e., Gaussain Markov random field)
- estimate Σ^{-1} via ℓ_1 regularized maximum likelihood

$$\text{minimize}_X \, \mathbf{Tr}(SX) - \log\det X + \lambda \|X\|_1$$

- ADMM form

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(SX) - \log\det X + \lambda\|Z\|_1 \\ \text{subject to} & X - Z = 0 \end{array}$$

- algorithm

$$\begin{aligned} X^{k+1} &:= \arg\min_X \mathbf{Tr}(SX) - \log\det X + (\rho/2)\|X - Z^k + U^k\|_F^2 \\ Z^{k+1} &:= S_{\lambda/\rho}(X^{k+1} + U^k/\rho) \\ U^{k+1} &:= U^k + \rho(X^{k+1} - Z^{k+1}) \end{aligned}$$

Analytical Solution for X-update

- first-order optimality condition

$$S - X^{-1} + \rho(X - Z^k + U^k) = 0$$

i.e.

$$\rho X - X^{-1} = \rho(Z^k - U^k) - S$$

- eigendecomposition $\rho(Z^k - U^k) - S = Q\Lambda Q^\intercal$
- form diagonal matrix $\tilde{X} = Q^\intercal X Q$ with

$$\tilde{X}_{ii} = \frac{\lambda_i + \sqrt{\lambda_i^2 + 4\rho}}{2\rho}$$

- let $X^{k+1} := Q\tilde{X}Q^\intercal$
- cost of X-update is an eigendecomposition

Table of Contents

Consensus optimization

- to solve problem with N objective terms

$$\text{minimize} \sum_{i=1}^{N} f_i(x)$$

e.g. f_i is the loss function for the ith block (mini-batch) of training data

- ADMM form

$$\begin{aligned} &\text{minimize} \sum_{i=1}^{N} f_i(x_i) \\ &\text{subject to} \quad x_i - z = 0 \end{aligned}$$

here

- x_is are local variables
- z is the global variable
- $x_i - z = 0$ are consistency or consensus constraints
- can add regularization using a $g(z)$ term.

Consensus optimization via ADMM

- $L_\rho(x, z, y) = \sum_{i=1}^{N}(f_i(x_i) + y_i^\mathsf{T}(x_i - z) + (\rho/2)\|x_i - z\|_2^2)$
- ADMM

$$
\begin{aligned}
x_i^{k+1} &:= \operatorname{arg\,min}_{x_i}(f_i(x_i) + \left(y_i^k\right)^\mathsf{T}(x_i - z^k) + (\rho/2)\|x_i - z^k\|_2^2 \\
z^{k+1} &:= \frac{1}{N}\sum_{i=1}^{N}(x_i^{k+1} + (1/\rho)y_i^k) \\
y_i^{k+1} &:= y_i^k + \rho(x_i^{k+1} - z^{k+1})
\end{aligned}
$$

- if regularization term included, averaging in z update is followed by $\mathbf{prox}_{g,\rho}$

The z-update can be written as

$$z^{k+1} = \bar{x}^{k+1} + \frac{1}{\rho}\bar{y}^k$$

Similarly, averaging the y-update, we have

$$\bar{y}^{k+1} = \bar{y}^k + \rho(\bar{x}^{k+1} - z^{k+1})$$

substituting z^{k+1} to $\bar{y}^{k+1}$ leads to $\bar{y}^{k+1} = 0$, which means

the dual variables have average value zero after the first iteration

Consensus optimization via ADMM

- using $\sum_i y_i^k = 0$, algorithm simplifies to

$$
\begin{aligned}
x_i^{k+1} &:= \arg\min_{x_i}\left(f_i(x_i) + \left(y_i^k\right)^T (x_i - \bar{x}^k) + (\rho/2)\|x_i - \bar{x}^k\|_2^2\right) \\
y_i^{k+1} &:= y_i^k + \rho(x_i^{k+1} - \bar{x}^{k+1})
\end{aligned}
$$

  where $\bar{x}^k = (1/N)\sum_i x_i^k$

- in each iteration
  - gather x_i^k and average to get $\bar{x}^k$
  - scatter the average $\bar{x}^k$ to processors
  - update y_i^k locally (in each processor, in parallel)
  - update x_i locally

Statistical interpretation

- f_i is negative log-likelihood for parameter x given ith data block
- x_i^{k+1} is an MAP estimate under prior $\mathcal{N}(\bar{x}^k + \frac{1}{\rho} y_i^k, \rho I)$
- prior mean is previous iteration's consensus shifted by 'price' of processor i disagreeing with previous consensus
- processors only need to support a Gaussian MAP method
    - type or number of data in each block not relevant
    - consensus protocol yields global maximum-likelihood estimate

Consensus classification

- data (examples) $(a_i, b_i), i = 1, \cdots, N, a_i \in \mathbb{R}^n, b_i \in \{+1, -1\}$
- linear classifier $\mathbf{sign}(a^T w + v)$, with weight w, offset v
- margin for ith example is $b_i(a_i^T w + v)$; want margin to be positive
- loss for ith example is $\ell(b_i(a_i^T w + v))$
  - ℓ is loss function, could be hinge, logistic, probit, exponential, etc...
- choose w, v to minimize

$$\frac{1}{N} \sum_{i=1}^{N} \ell(b_i(a_i^T w + v)) + r(w)$$

- split data and use ADMM consensus to solve

In case of SVM with hinge loss and ℓ_2-regularization, the ADMM algorithm

$$
\begin{aligned}
x_i^{k+1} &= \arg\min_{x_i}\left(\mathbf{1}^\intercal(A_i x_i + \mathbf{1})_+ + \frac{\rho}{2}\|x_i - z^k + u_i^k\|_2^2\right) \\
z^{k+1} &= \frac{\rho}{(1/\lambda) + N\rho}(\bar{x}^{k+1} + \bar{u}^k) \\
u_i^{k+1} &= u_i^k + x_i^{k+1} - z^{k+1}
\end{aligned}
$$

Interpretation

- each x_i-update involves fitting a SVM to local data A_i with an offset in the regularization term
- the dual variable z gathers the solutions for consensus
- the dual variable u update the offset

Consensus SVM example

- hinge loss $\ell(u) = (1 - u)_+$ with ℓ_2 regularization
- toy problem with $n = 2$, $N = 400$ to illustrate
- examples split into 20 groups, in worst possible way: each group contains only positive or negative examples

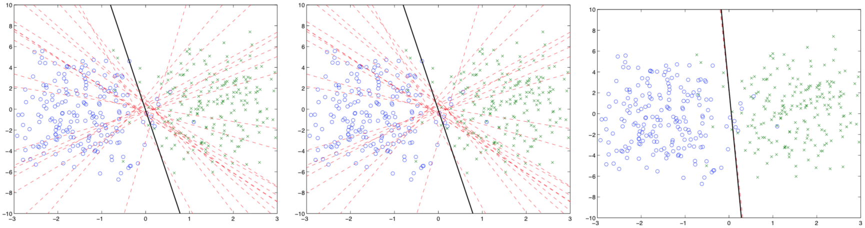

Figure: training iterations $1, 5, 40$

Table of Contents

- ADMM gives simple single-processor algorithms that can be competitive with state-of-the-art
- can be used to coordinate many processors, each solving a substantial problem, to solve a very large problem

Table of Contents

- big data techniques
  - computational statistics, machine learning
  - especially on large data sets
    - data fusion
    - heterogeneous and homogeneous data sets
    - stream data
    - small data learning
- optimization
  - loss function — data associated, summation form, task-specific, determined by data modelling