Convex Optimization II

Lecture 19: Alternating Direction Method of Multipliers

Hamed Shah-Mansouri

Department of Electrical Engineering

Sharif University of Technology

1400-2

Motivations

To robustify and expedite the Lagrangian multiplier method for

- Large-scale optimization
  - machine learning/statistics/signal processing with huge datasets, dynamic optimization on large-scale networks
- Decentralized optimization
  - distributed storage, big data.

Thanks to Prof. Stephen Boyd and Prof. Dimitri Bertsekas for part of the slides used in this lecture.

REFERENCES

[1] D.P. Bertsekas, “Multiplier methods: A survey,” Automatica, vol.12, no.2, pp.133-145, March 1976.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein,“Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers,” *Found. Trends Mach. Learn*, vol. 3, no. 1, pp. 1-120, Jan. 2011.

[3] M. Leinonen, M. Codreanu, and M. Juntti, “Distributed Joint Resource and Routing Optimization in Wireless Sensor Networks via Alternating Direction Method of Multipliers,” *IEEE Trans. on Wireless Communications*, vol. 12, no. 11, pp. 5454-5467, Nov. 2013.

Lagrangian Multiplier

- Recall from the previous lectures that, for a convex equality constrained optimization problem ($f : \mathbf{R}^n \to \mathbf{R}$)

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}}\ f(\mathbf{x}) \\ &\text{subject to } A\mathbf{x} - b = \mathbf{0}. \end{aligned}$$

- Lagrangian

$$L(\mathbf{x}, \boldsymbol{\nu}) = f(\mathbf{x}) + \boldsymbol{\nu}^T (A\mathbf{x} - b).$$

- Dual function

$$g(\boldsymbol{\nu}) = \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\nu}).$$

- Dual problem

$$\underset{\boldsymbol{\nu}}{\text{maximize}}\ g(\boldsymbol{\nu}).$$

Lagrangian Multiplier

Gradient method for dual problem (iteration k)

- Primal minimization
$$\mathbf{x}^{k+1} := \operatorname*{argmin}_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\nu}^k).$$
- Dual update
$$\boldsymbol{\nu}^{k+1} := \boldsymbol{\nu}^k + \alpha^k (A\mathbf{x}^{k+1} - b).$$

The gradient method converges to the optimal solution

- under some strong assumptions (e.g., convexity, differentiability)
- often slow.

Dual Decomposition

- Suppose $f(\mathbf{x})$ is separable in $\mathbf{x} = (x_1, \ldots, x_N)$

$$f(\mathbf{x}) = f_1(x_1) + \ldots + f_N(x_N)$$

- Then L is separable in $\mathbf{x}$

$$\begin{aligned} L(\mathbf{x}, \boldsymbol{\nu}) &= L_1(x_1, \boldsymbol{\nu}) + \ldots + L_N(x_N, \boldsymbol{\nu}) - \boldsymbol{\nu}^T b \\ L_i(x_i, \boldsymbol{\nu}) &= f_i(x_i) + \nu^T A_i x_i, \qquad i = 1, \ldots, N. \end{aligned}$$

- Primal minimization splits into N separate problems (can be run in parallel)

$$x_i^{k+1} := \underset{x_i}{\operatorname{argmin}}\, L_i(x_i, \boldsymbol{\nu}^k).$$

- Dual update

$$\boldsymbol{\nu}^{k+1} := \boldsymbol{\nu}^k + \alpha^k (A\mathbf{x}^{k+1} - b).$$

History

- Penalty methods are a certain class of algorithms for solving constrained optimization problems.
- Augmented Lagrangian methods have similarities to penalty methods. It adds another penalty term, designed to mimic a Lagrange multiplier.
- The method was originally known as the method of multipliers , and was studied much in the 1970 and 1980s as a good alternative to penalty methods (*Hestenes, Powell*; analysis by *Bertsekas* 1982).
- Alternating Direction Method of Multipliers (ADMM) is a method of multipliers to robustify Lagrangian multipliers method and support decomposition (*Gabay, Mercier, Glowinski, Marrocco* 1976).

Penalty Methods

- Consider the equality constrained problem

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & h_i(\mathbf{x}) = 0, \quad i = 1, \ldots, m \\ & \mathbf{x} \in \mathcal{X}, \end{array}$$

where $f : \mathbf{R}^n \to \mathbf{R}$ and $h_i : \mathbf{R}^n \to \mathbf{R}$ are continuous.

- This problem can be solved as a series of unconstrained minimization problems

$$\text{minimize} \quad f(\mathbf{x}) + c_k \sum_{i=1}^{m} \phi\left(h_i(\mathbf{x})\right),$$

for a scalar sequence $\{c_k\}$ such that $0 < c_k \leq c_{k+1}$ and $c_k \to \infty$.

- The penalty function $\phi : \mathbf{R} \to \mathbf{R}_+$ is such that

$$\phi(t) \geq 0, \ \forall t, \quad \phi(t) = 0 \ \text{ if and only if } t = 0.$$

Penalty Methods

- The most common penalty function is the quadratic function

$$\phi(t) = \frac{1}{2}t^2.$$

- The optimal value of the original problem can be obtained from

$$\inf_{x \in \mathcal{X}} \lim_{k \to \infty} \left\{ f(\mathbf{x}) + c_k \sum_{i=1}^{m} \phi\left(h_i(\mathbf{x})\right) \right\}.$$

- Penalty methods are simple to implement, are applicable to a broad class of problems and take advantage of the very powerful unconstrained minimization methods.

- On the negative side, penalty methods usually suffer from slow convergence and numerical instabilities associated with ill-conditioning induced by large values of the penalty parameter c_k.

Penalty Methods – Example

Consider the following problem

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}}\ f(\mathbf{x}) = x_1^2 + x_2^2 \\ &\text{subject to } x_1 = 1. \end{aligned}$$

The above constrained problem can be solved as a series of unconstrained minimization problems with the following objective function.

$$x_1^2 + x_2^2 + \frac{c_k}{2}(x_1 - 1)^2.$$

METHOD OF MULTIPLIERS

- a method to mitigate the drawbacks of penalty methods.
- a method to robustify and expedite the Lagrangian multipliers method.
- Consider the equality constrained optimization problem

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}}\ f(\mathbf{x}) \\ &\text{subject to } A\mathbf{x} - b = \mathbf{0}. \end{aligned}$$

- use augmented Lagrangian (*Hestenes, Powell* 1969)

$$L_\rho(\mathbf{x}, \boldsymbol{\nu}) = f(\mathbf{x}) + \boldsymbol{\nu}^T(A\mathbf{x} - b) + \frac{\rho}{2}\|(A\mathbf{x} - b)\|_2^2,$$

where ρ is finite.

Method of Multipliers

- method of multipliers (*Hestenes, Powell*; analysis by *Bertsekas* 1982) consists of the iterations

$$\mathbf{x}^{k+1} := \underset{\mathbf{x}}{\operatorname{argmin}}\, L_\rho(\mathbf{x}, \boldsymbol{\nu}^k).$$

$$\boldsymbol{\nu}^{k+1} := \boldsymbol{\nu}^k + \rho(A\mathbf{x}^{k+1} - b).$$

- faster convergence under much more relaxed conditions (f can be non-differentiable, take on value ∞, ρ can be large)
- Question: Can we still use the decomposition method?

Alternating Direction Method of Multipliers

- ADMM, proposed by *Gabay, Mercier, Glowinski, Marrocco* in 1976
  - achieves the good robustness of the method of multipliers
  - and supports decomposition

- Problem formulation

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}}\ & f_1(x_1) + \ldots + f_N(x_N) \\ \text{subject to}\ & A\mathbf{x} - b = \mathbf{0}. \end{aligned}$$

- Augmented Lagrangian

$$L_\rho(\mathbf{x}, \boldsymbol{\nu}) = f_1(x_1) + \ldots + f_N(x_N) + \boldsymbol{\nu}^T(A\mathbf{x} - b) + \frac{\rho}{2}\|(A\mathbf{x} - b)\|_2^2.$$

- ADMM

$$x_i^{k+1} := \underset{x_i}{\text{argmin}}\ L_\rho(\mathbf{x}, \boldsymbol{\nu}^k).$$

$$\boldsymbol{\nu}^{k+1} := \boldsymbol{\nu}^k + \rho(A\mathbf{x}^{k+1} - b).$$

Alternating Direction Method of Multipliers

- Primal optimization: minimize over x_i assuming fixed $x_j, j = 1, \ldots, N, i \neq j$
- If we minimize over $x_i, i = 1, \ldots, N$ jointly, reduces to method of multipliers
- Convergence condition (a sufficient condition)
  - The extended real function $f \ : \ \mathbf{R}^n \to \mathbf{R} \cup \{+\infty\}$ is closed, proper, and convex.
- Practical examples show that ADMM can be very slow to converge to high accuracy. However, it is often the case that ADMM converges to modest accuracy, sufficient for many applications, within a few tens of iterations.
  Question: Why?

EXAMPLE

Solve a problem with N objective terms

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

Application: Large-scale convex optimization

ADMM WITH SCALED DUAL VARIABLES

- Combine linear and quadratic terms in augmented Lagrangian

$$
\begin{aligned}
L_\rho(\mathbf{x}, \boldsymbol{\nu}) &= f(\mathbf{x}) + \boldsymbol{\nu}^T(A\mathbf{x} - b) + \frac{\rho}{2}\|(A\mathbf{x} - b)\|_2^2 \\
&= f(\mathbf{x}) + \frac{\rho}{2}\|(A\mathbf{x} - b + u)\|_2^2 + \text{const.}
\end{aligned}
$$

with $u = (1/\rho)\ \nu$

- ADMM (scaled dual form)

$$
x_i^{k+1} := \underset{x_i}{\operatorname{argmin}}\ f(\mathbf{x}) + \frac{\rho}{2}\|(A\mathbf{x} - b + u)\|_2^2.
$$

$$
\boldsymbol{u}^{k+1} := \boldsymbol{u}^k + A\mathbf{x}^{k+1} - b.
$$

Application of ADMM in WSN

[3] M. Leinonen, M. Codreanu, and M. Juntti, "Distributed Joint Resource and Routing Optimization in Wireless Sensor Networks via Alternating Direction Method of Multipliers," *IEEE Trans. on Wireless Communications*, vol. 12, no. 11, pp. 5454-5467, Nov. 2013.

Distributed total transmit power minimization in a multi-hop single-sink data gathering wireless sensor network.

- Let $\mathcal{N}$ and $\mathcal{L}$ denote the set of nodes and links, respectively.
- f_l denotes the amount of flow in each link $l \in \mathcal{L}$, and $\boldsymbol{f} = (f_1, \ldots, f_L)^T$.
- Each sensor node i is associated with a fixed external flow $r_i > 0$, that is the source rate of node i.

Application of ADMM in WSN

- Key idea: conservation flow constraint

$$\sum_{l \in \mathcal{O}(i)} f_l - \sum_{l \in \mathcal{I}_l} f_l = r_i, \forall i \in \mathcal{N},$$

$$\boldsymbol{A}\boldsymbol{f} = \boldsymbol{r}.$$

- Problem formulation

$$\begin{array}{ll}
\text{minimize} & \sum_{l \in \mathcal{L}} p_l \\
\text{subject to} & \boldsymbol{A}\boldsymbol{f} = \boldsymbol{r} \\
& f_l \leq w_l \log\left(1 + \dfrac{\gamma_l p_l}{w_l}\right), \ \forall l \in \mathcal{L} \\
& \sum_{l \in \mathcal{O}(i)} p_l \leq P, \ \forall i \in \mathcal{S} \\
& \sum_{l \in \mathcal{O}(i)} w_l \leq W_i, \ \forall i \in \mathcal{S} \\
& p_l \geq 0, \ w_l \geq 0, \ f_l \geq 0, \ \forall l \in \mathcal{L},
\end{array}$$

Application of ADMM in WSN

Two different algorithms to solve the above problem

- Dual decomposition
- ADMM

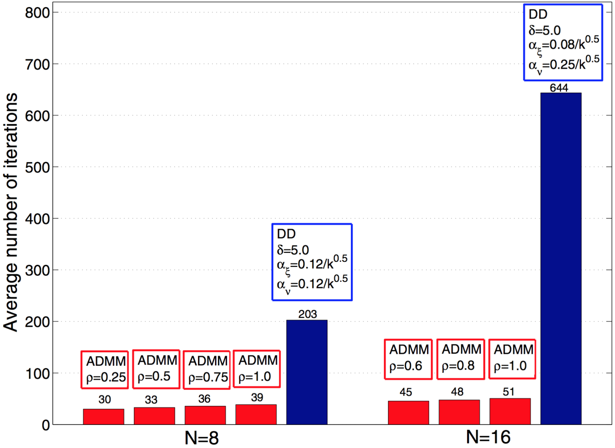

Figure: Average number of iterations [3]

SUMMARY

- Lagrangian multipliers method
  - works under some strong assumptions
  - often slow
- Method of Multipliers
  - is robust
  - but cannot support decomposition
- ADMM
  - is robust
  - and supports decomposition