ROYAL INSTITUTE
OF TECHNOLOGY

EP3260: Machine Learning Over Networks

Lecture 7: Alternating Direction Method of Multipliers

José Mairton B. da Silva Jr.

Division of Network and Systems Engineering
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology, Stockholm, Sweden

https://sites.google.com/view/mlons2020/home

April 2020

Learning outcomes

- Basics of dual decomposition and method of multipliers
- Fundamentals of ADMM
- Convergence analysis with hyperparameter optimization
- Applications on consensus optimization

Outline

1. Preliminaries

2. ADMM

3. Application examples

4. References

Outline

1. Preliminaries
   - Dual decomposition
   - Method of multipliers

2. ADMM

3. Application examples

4. References

Dual problem

Convex problem

$$\begin{array}{rl} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array} \tag{1}$$

Lagrangian function

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

where y is the dual variable (Lagrange multiplier)

Dual problem is defined as

$$\text{maximize} \quad \{g(y) = \inf_x L(x, y)\}$$

Recover a primal optimal $x^\star$ from a dual optimal $y^\star$

$$x^\star = \text{argmin}_x \quad L(x, y^\star)$$

Dual ascent

- Perform an x-minimization step

$$x^{k+1} = \operatorname*{argmin}_x L(x, y^k)$$

- Calculate $\nabla g(y) = Ax^{k+1} - b$ (assuming g is differentiable), then update the dual variable using gradient method

$$y^{k+1} = y^k + \alpha^k (Ax^{k+1} - b)$$

- More properties
  - Under proper choice of step-size α^k and other conditions (e.g. KKT conditions), (x^k, y^k) converges to an optimal primal-dual pair.
  - Dual ascent because, under proper choice of α^k, the dual function converges monotonically $g(y^{k+1}) > g(y^k)$.
  - If g is non-differentiable then $Ax^{k+1} - b$ is a sub-gradient of $-g$ and convergence is non-monotonic.

Dual decomposition

Suppose $f(x) = f_1(x_1) + \cdots + f_N(x_N)$, $x = (x_1, \ldots, x_N)$, with f_i and x_i corresponding to a partition of the original problem

$$\begin{aligned} \text{minimize } & f(x) = \sum_{i=1}^{N} f_i(x_i) \\ \text{subject to } & \sum_{i=1}^{N} A_i x_i = b \end{aligned}$$

The Lagrangian is also separable (in x)

$$L(x,y) = \sum_{i=1}^{N} L_i(x_i, y) = \sum_{i=1}^{N} f_i(x_i) + \sum_{i=1}^{N} y^\top A_i x_i - y^\top b$$

The x-minimization step in dual ascent method splits into N parallel minimization

$$x_i^{k+1} = \underset{x_i}{\operatorname{argmin}}\, L_i(x_i, y^k)$$

Dual decomposition

The dual decomposition method (dates back to 60's)

$$x_i^{k+1} = \operatorname*{argmin}_{x_i} L_i(x_i, y^k), \quad i = 1, \ldots, N,$$

$$y^{k+1} = y^k + \alpha^k (\sum_{i=1}^{N} A_i x_i^{k+1} - b)$$

A potentially large-scale problem is solved iteratively in a distributed fashion

- Perform parallel independent x_i-minimization step
- Gather residuals $A_i x_i^{k+1}$ to calculate global dual variable y^{k+1}; then broadcast it to distributed workers

Under several assumptions, it converges; however, often slowly!

Method of multipliers

Augmented Lagrangian (with penalty parameter $\rho > 0$)

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

It can be viewed as Lagrangian of the problem

$$\begin{aligned} \text{minimize } & f(x) + \frac{\rho}{2}\|Ax - b\|_2^2 \\ \text{subject to } & Ax = b \end{aligned}$$

which is equivalent to original problem.
Forming the associated dual function $g_\rho(y) = \inf_x L_\rho(x,y)$ and applying dual ascent

$$\begin{aligned} x^{k+1} &= \underset{x}{\operatorname{argmin}}\, L_\rho(x, y^k), \\ y^{k+1} &= y^k + \rho(Ax^{k+1} - b) \end{aligned}$$

The x-minimization step uses the augmented Lagrangian, and the penalty parameter ρ is used as the step size α^k.

Method of multipliers

It converges under more general conditions than dual ascent, including cases when f is unbounded from above or it is not strictly convex.

Under primal-dual feasibility (i.e., the optimality conditions of problem (1))

$$Ax^\star - b = 0, \quad \nabla f(x^\star) + A^\top y^\star = 0,$$

Since x^{k+1} minimizes $L_\rho(x, y^k)$

$$\begin{aligned} 0 &= \nabla_x L_\rho(x^{k+1}, y^k) \\ &= \nabla f(x^{k+1}) + A^\top (y^k + \rho(Ax^{k+1} - b)) \\ &= \nabla f(x^{k+1}) + A^\top y^{k+1} \end{aligned}$$

Particular step-size choice $\alpha^k = \rho$ at dual-update ($y^{k+1} = y^k + \rho(Ax^{k+1} - b)$) makes (x^{k+1}, y^k) dual feasible!

Achieve primal optimality at the limit: $Ax^{k+1} - b \to 0$.

Method of multipliers

- Method of multiplier enjoys greatly improved convergence properties compared to dual-ascent.
- However, it comes at a cost.

Method of multipliers

- Method of multiplier enjoys greatly improved convergence properties compared to dual-ascent.
- However, it comes at a cost.
- When f is separable, the augmented Lagrangian L_ρ is not separable, so the x-minimization step cannot be carried out separately in parallel for x_i's.
- Next, we will study a method to solve this issue!

Outline

1. Preliminaries

2. ADMM
   - Alternating direction method of multipliers
   - Proximal operators
   - Convergence
   - Hyper-parameters optimization

3. Application examples

4. References

Alternating direction method of multipliers

- Introduced in the 70's and revisited for a variety of applications
  - Decentralized and large-scale problems while imposing little communication overhead
  - Machine learning and statistical problems with huge data sizes
- Improved convergence properties of method of multiplier, and effective heuristic methods in many non-convex problems
- Solves the issue of non-decomposability of the augmented Lagrangian L_ρ
- Performs Gauss-Seidel decomposition in the primal update so the x-minimization step can be carried out separately in parallel for x_i's.

Alternating direction method of multipliers

ADMM canonical form: convex f and g with two sets of variables and separable costs

$$\begin{aligned} \text{minimize } & f(x) + g(z) \\ \text{subject to } & Ax + Bz = c \end{aligned} \tag{2}$$

Augmented Lagrangian:

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

ADMM iterates:

$$\begin{aligned} x^{k+1} &= \operatorname*{argmin}_x L_\rho(x, z^k, y^k), \\ z^{k+1} &= \operatorname*{argmin}_z L_\rho(x^{k+1}, z, y^k), \\ y^{k+1} &= y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \end{aligned}$$

Remarks

- Minimizing jointly over x and z falls back to method of multipliers!
- Once minimizing over x or z, the other variable is kept fixed (one pass of Gauss-Seidel)
- Recall from decomposition method: if problem variables are separable

$$\begin{aligned} \text{minimize } & f(x) = \textstyle\sum_{i\in[N]} f_i(x_i) \\ \text{subject to } & \textstyle\sum_{i\in[N]} A_i x_i = b \end{aligned}$$

then, $L_\rho(x,y) = \sum_{i=1}^N f_i(x_i) + \sum_{i=1}^N y^\top A_i x_i - y^\top b + \dfrac{\rho}{2}\|\sum_{i=1}^N A_i x_i - b\|_2^2$

- Applying ADMM, now we get separable x_i-updates

$$\begin{aligned} x_i^{k+1} &= \operatorname*{argmin}_{x_i} L_\rho(x_i, x_j^k, y^k), \\ &= \operatorname*{argmin}_{x_i} f_i(x_i) + y^{k\top} A_i x_i + \frac{\rho}{2}\|A_i x_i - b\|_2^2 + \rho x_i^\top A_i^\top \sum_{i\neq j} A_j x_j^k. \end{aligned}$$

Optimality conditions

- Primal-dual feasibility (i.e., the optimality conditions of problem (2))

$$Ax^\star + Bz^\star = c, \quad \nabla f(x^\star) + A^\top y^\star = 0, \quad \nabla g(z^\star) + B^\top y^\star = 0$$

- Since z^{k+1} minimizes $L_\rho(x^{k+1}, z, y^k)$, one has

$$\begin{aligned} 0 &= \nabla_z L_\rho(x^{k+1}, z, y^k) \\ &= \nabla g(z^{k+1}) + B^\top(y^k + \rho(Ax^{k+1} + Bz^{k+1} - c)) \\ &= \nabla g(z^{k+1}) + B^\top y^{k+1} \end{aligned}$$

- Particular step-size choice $\alpha^k = \rho$ at dual-update ($y^{k+1} = y^k + \rho(Ax^{k+1} + Bz^{k+1} - c)$) satisfies second dual feasibility condition!

- Achieve primal and first dual feasibility at the limit

$$Ax^k + Bz^k - c \to 0, \quad \nabla f(x^k) + A^\top y^k \to 0$$

ADMM with scaled dual variables

- Combine linear and quadratic terms in augmented Lagrangian

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

with $u = (1/\rho)y^k$.

- ADMM iterates (in scaled form):

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

Proximal operators

- Consider x-update (when $A = I$)

$$x^{k+1} = \underset{x}{\text{argmin}}\ f(x) + \frac{\rho}{2}\|x + \underbrace{Bz^k - c + u^k}_{\text{const.} \triangleq -v}\|_2^2,$$

$$= \underset{x}{\text{argmin}}\ f(x) + \frac{\rho}{2}\|x - v\|_2^2 = \text{prox}_{f,\rho}(v),$$

- Example 1: indicator function of set C: $f = I_C$. Then $x^{k+1} = \Pi_C(v)$, or projection onto C.

$$f(x) = I_{\geq b}(x) = \begin{cases} 0 & \text{if } x \geq b, \\ \infty & \text{otherwise.} \end{cases} \quad \text{then } x^{k+1} = \max(v, b).$$

- Example 2: ℓ_1-norm (Lasso problem): minimize $(1/2)\|Ax - b\|_2^2 + \lambda\|x\|_1$. Formulating the ADMM form, resulting z-update is called soft thresholding: $z_i^{k+1} = S_{\lambda/\rho}(x_i^{k+1} + u_i^k)$,

$$s_a(v) = \begin{cases} v - a & v > a \\ 0 & |v| \leq a \\ v + a & v < -a \end{cases}$$

Convergence

General case holds under two assumptions:

(1) The extended real valued functions $f \in \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $g \in \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ are closed, proper and convex.

- Function f satisfies the assumption iff its epigraph

$$\textsf{epi}\, f = \{(x,t) \in \mathbb{R}^n \times \mathbb{R} |\, f(x) \leq t\}$$

  is a closed nonempty convex set.
- There are x and z (not necessarily unique) that minimize the augmented Lagrangian (sub-problems in x and z-update are solvable).

(2) The unaugmented Lagrangian has a saddle point: there exist $(x^\star, z^\star, y^\star)$ such that the following holds;

$$L_0(x^\star, z^\star, y) \leq L_0(x^\star, z^\star, y^\star) \leq L_0(x, z, y^\star) \quad \forall x, z, y.$$

Assumption 1 and 2 imply that strongly duality holds; no explicit assumptions on A and B.

Convergence

Under Assumptions 1 and 2, the ADMM iterates satisfy:

- Residual convergence (primal): $r^k = Ax^k + Bz^k - c \to 0$ as $k \to \infty$.
- Objective convergence $f(x^k) + g(z^k) \to f(x^\star) + g(z^\star)$ as $k \to \infty$.
- Dual variable convergence: $y^k \to y^\star$ as $k \to \infty$.
- Note that (under these assumptions only) x^k and z^k do not necessarily converge to optimal values.

In practice, it is possible to construct examples where ADMM converges very slow. However, often it is easy to tune ADMM to converge to modest accuracy after few tens of iterations.

Optimal step-size

For standard QP with positive definite Hessian

$$\begin{array}{ll}\text{minimize} & \frac{1}{2}x^\top Qx + q^\top x \\ \text{subject to} & Ax \leq c,\end{array} \rightarrow \begin{array}{ll}\text{minimize} & \frac{1}{2}x^\top Qx + q^\top x + I_+(z), \\ \text{subject to} & Ax + z = c,\end{array}$$

The ADMM iterates converge linearly: in terms of some residuals, e.g., $\|r^k\|_2 \leq \delta\gamma^k\|r^0\|_2$ for some $\delta \in \mathbb{R}_+, \gamma \in (0,1)$

Moreover, the optimum choice of step-size ρ and corresponding convergence factor can be found as

$$\rho^\star = \frac{1}{\sqrt{\lambda_{\min}\lambda_{\max}}}, \quad \gamma^\star = \frac{\kappa - 1}{\kappa + 1}, \quad \kappa = \frac{\lambda_{\max}}{\lambda_{\min}}, \quad \lambda_i \triangleq \lambda(AQ^{-1}A^\top).$$

Although the linear rate is established for general QPs, the optimal parameter holds when A is full-row rank (or invertible).

Optimal step-size

The result can be generalized for full-row rank A and L-smooth and μ-strongly convex f!

- A convex f is L-smooth and μ-strongly convex if these two are also convex
$$\frac{L}{2}\|x\|_2^2 - f(x), \quad f(x) - \frac{\mu}{2}\|x\|_2^2$$
- The ADMM iterates converge linearly with optimal step-size and convergence factor

$$\rho^\star = \sqrt{\frac{L\mu}{\lambda_{\min}(AA^\top)\lambda_{\max}(AA^\top)}}, \quad \gamma^\star = \frac{\bar{\kappa}-1}{\bar{\kappa}+1}, \quad \bar{\kappa} = \frac{L\lambda_{\max(}AA^\top)}{\mu\lambda_{\min}(AA^\top)}$$

Other ways of parameter tuning

- Relaxation: For z and y-updates, the term Ax^{k+1} can be replaced with

$$\alpha^k A x^{k+1} - (1-\alpha^k)(Bz^k - c), \quad \alpha^k \in (0,2)$$

  - For $\alpha^k > 1$ it is called over-relaxation, and for $\alpha^k < 1$ is under-relaxation
  - Theoretical best value: $\alpha^\star = 2$ for L-smooth and μ-strongly convex f.
  - Empirical results suggest $\alpha^k \in [1.5, 1.8]$ for improved convergence!

- Varying step-size ρ^k: empirically, convergence improvement can be achieved by changing ρ^k at each iteration to balance the primal and the dual residuals (not for cases with analytical $\rho^\star$).

$$\rho^{k+1} = \begin{cases} \tau^{\text{incr}} \rho^k & \text{if } \dfrac{\|r^k\|_2}{\|s^k\|_2} > \delta \\ \tau^{\text{decr}} \rho^k & \text{if } \dfrac{\|s^k\|_2}{\|r^k\|_2} > \delta, \\ \rho^k & \text{otherwise.} \end{cases}, \qquad \begin{aligned} r^k &= Ax^k + Bz^k - c \\ s^k &= \rho^k A^\top B(z^k - z^{k-1}) \end{aligned}$$

where $\delta > 0, \tau^{\text{incr}} > 1, \tau^{\text{decr}} < 1$. Typical example $\delta = 10$, $\tau^{\text{incr}} = 2, \tau^{\text{decr}} = 0.5$.

Outline

1. Preliminaries

2. ADMM

3. Application examples

4. References

Consensus optimization

Problem with N objective terms: minimize $\sum_{i=1}^N f_i(x)$;

f_i might be the loss function for i-th block of training data $\{(a_i, b_i)\}_{i=1}^N$ in a statistical optimization problem: e.g.,

- LASSO:

$$\text{minimize}_{x\in\mathbb{R}^m} \quad \frac{1}{N}\sum_{i=1}^N (b_i - x^\top a_i)^2 + \lambda\|x\|_1$$

- Classification: $a_i \in \mathbb{R}^{m_i}, b \in \{-1, +1\}$, $l(\cdot)$ a loss function (hinge, logistic, etc), $r(\cdot)$ a regularization function $(\ell_1, \ell_2, \dots)$

$$\text{minimize}_{x\in\mathbb{R}^m, w\in\mathbb{R}} \quad \frac{1}{N}\sum_{i=1}^N l(b_i(a_i^\top x + w)) + r(x)$$

- Support vector machine (SVM):

$$\text{minimize}_{x\in\mathbb{R}^m, w\in\mathbb{R}} \quad \frac{1}{N}\sum_{i=1}^N \max\{0, 1 - b_i(x^\top a_i + w)\} + \lambda\|x\|_2^2$$

Consensus optimization

Problem with N objective terms: minimize $\sum_{i=1}^{N} f_i(x)$;

ADMM iterates:

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

- x_i are local variables
- z is global variable (kept in a central node)
- $x_i - z = 0$ are consistency or *consensus* variables
- can add regularization by adding $g(z)$ term into objective

Consensus optimization- ADMM formulation

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

ADMM form:

$$x_i^{k+1} = \underset{x_i}{\text{argmin}} \left(f_i(x_i) + y_i^{k\top}(x_i - z^k) + \rho/2\|x_i - z^k\|_2^2 \right)$$

$$z^{k+1} = \frac{1}{N}\sum_{i=1}^N \left(x_i^{k+1} + 1/\rho y_i^k \right)$$

$$y_i^{k+1} = y_i^k + \rho(x_i^{k+1} - z^{k+1})$$

Under regularization, the averaging term in z-update is replaced with $\text{prox}_{g,\rho}$.

Consensus optimization- ADMM formulation

One can check $\sum_{i=1}^N y_i^k = 0$, which further simplifies the ADMM algorithm to

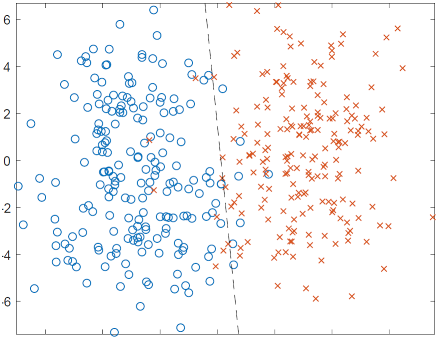
$$
\begin{aligned}
x_i^{k+1} &= \underset{x_i}{\operatorname{argmin}} \left(f_i(x_i) + y_i^{k\top}(x_i - \bar{x}^k) + \rho/2 \|x_i - \bar{x}^k\|_2^2 \right) \\
y_i^{k+1} &= y_i^k + \rho(x_i^{k+1} - \bar{x}^{k+1})
\end{aligned}
$$

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

At each iteration

- local processors compute y_i^k and x_i^k in parallel using global variable $\bar{x}^{k-1}$
- central node gathers x_i^k from local processors and computes the average $\bar{x}^k$
- central node scatters the average $\bar{x}^k$ to processors

Performance of ADMM- Consensus SVM



- Problem formulation:

  minimize$_{x\in\mathbb{R}^m, w\in\mathbb{R}}$ $\frac{1}{N}\sum_{i=1}^N \max\{0, 1-b_i(x^\top a_i + w)\} + \lambda\|x\|_2^2$

- Example with $a_i \in \mathbb{R}^{2\times 400}$ samples partitioned in the worst way (2 subsystems that each holds only positive $b_i = +1$ or negative $b_i = -1$ examples)

- After 60 iterations of consensus ADMM, primal and dual residuals decay to 10^{-2}

Networked consensus optimization

Problem with N objective terms: minimize $\sum_{i=1}^{N} f_i(x)$;

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{N} f_i(x_i) \\ \text{subject to} & x_i = z_{ij}, \quad \text{for } i = 1, \ldots, N,\ j \in \mathcal{N}_i, \\ & z_{ij} = z_{ji}, \quad \text{for } (i,j) \in \mathcal{E}. \end{array}$$

- x_i are local variables
- z_{ij} are auxiliary edge variables (kept locally)
- There are other ways to formulate consensus variables (node formulation, etc.)

Networked consensus optimization - ADMM

$$L_\rho(x, y, z) =$$
$$\textstyle\sum_{i=1}^N f_i(x_i) + \left(\sum_{i=1}^N \sum_{j \in \mathcal{N}_i} y_{ij}^\top (x_i - z_{ij}) + \frac{\rho}{2} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|x_i - z_{ij}\|_2^2\right)$$

ADMM form:

$$\begin{aligned}
x_i^{k+1} &= \underset{x_i}{\text{argmin}} \left(f_i(x_i) + \sum_{j \in \mathcal{N}_i} y_{ij}^{k\top}(x_i - z_{ij}^k) + \frac{\rho}{2} \sum_{j \in \mathcal{N}_i} \|x_i - z_{ij}^k\|_2^2 \right) \\
z_{ij}^{k+1} &= \frac{1}{2}\left(x_i^{k+1} + x_j^{k+1} + 1/\rho(y_{ij}^k + y_{ji}^k)\right) \\
y_{ij}^{k+1} &= y_{ij}^k + \rho(x_i^{k+1} - z_{ij}^{k+1})
\end{aligned}$$

Under regularization, the averaging term in z_{ij}-update is replaced with $\text{prox}_{g,\rho}$ of an average point!

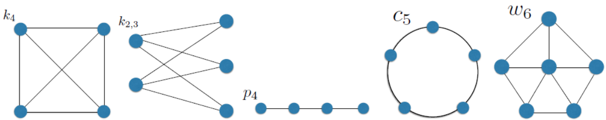
Networked consensus - Topology-dependent

After some algebra tricks, simplified decentralized ADMM form (with $g(z) = 0$)

$$\partial f(x_i^{k+1}) + y_i^k + 2\rho|\mathcal{N}_i|x_i^{k+1} - \rho\left(|\mathcal{N}_i|x_i^k + \sum_{j\in\mathcal{N}_i} x_j^k\right) = 0,$$

$$y_i^{k+1} = y_i^k + \rho\left(|\mathcal{N}_i|x_i^{k+1} - \sum_{j\in\mathcal{N}_i} x_j^{k+1}\right).$$

- $y_i \in \mathbb{R}^{m_i}, i = 1, \ldots, N$ is the local Lagrange multiplier at agent i
- Performance of decentralized ADMM also depends on underlying network topology (Shi&Ling, 2013; Teixeira&Ghadimi, 2016;Makhdoumi&Ozdaglar, 2017)



CA 5: ADMM

Split the "MNIST" dataset to 10 random disjoint subsets, each for one worker, and consider SVM classifier in the form of $\min_{\boldsymbol{w}} \frac{1}{N} \sum_{i \in [N]} f_i(\boldsymbol{w})$ with $N = 10$.

a) Run decentralized GD (from Lecture 6) with 10 workers. Characterize the convergence against the total number of signaling exchanges among all nodes, denoted by T.

b) Consider a two-star topology with communication graph (1,2,3,4)-5-6-(7,8,9,10) and run decentralized subgradient method (from Lecture 6) and ADMM over network (from Lecture 7). Characterize the convergence against T. Tune hyperparameters to improve the convergence rate.

c) Propose an approach to reduce T with a marginal impact on the convergence. Do not limit your imaginations and feel free to propose any change or any solution. While being nonsense in some applications, your solution may actually make very sense in some other applications. Discuss pros and cons of your solution and possibly provide numerical evidence that it reduces T.

Outline

1. Preliminaries
2. ADMM
3. Application examples
4. References

Some references

- S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” FoT in Machine Learning, 2011.

- E. Ghadimi, A. Teixeira, I. Shames, and M. johansson, “Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems,” IEEE Transactions on Automatic Control, 2015.

- P. Giselsson, and S. Boyd, “Linear convergence and metric selection for Douglas-Rachford splitting and ADMM,” IEEE Transactions on Automatic Control, 2017.

- W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, “On the linear convergence of the ADMM in decentralized consensus optimization,” IEEE Transactions on Signal Processing, 2014.

- A. Teixeira, E. Ghadimi, I. Shames, H. Sandberg, and M. Johansson, “The ADMM algorithm for distributed quadratic problems: Parameter selection and constraint preconditioning,” IEEE Transactions on Signal Processing. 2016.

- A. Makhdoumi, and A. Ozdaglar, “Convergence rate of distributed ADMM over networks,” IEEE Transactions on Automatic Control, 2017.

ROYAL INSTITUTE
OF TECHNOLOGY

EP3260: Machine Learning Over Networks

Lecture 7: Alternating Direction Method of Multipliers

José Mairton B. da Silva Jr.

Division of Network and Systems Engineering
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology, Stockholm, Sweden

`https://sites.google.com/view/mlons2020/home`

April 2020