On the $O(1/k)$ Convergence of Asynchronous Distributed Alternating Direction Method of Multipliers

Ermin Wei Asu Ozdaglar

Electrical Engineering and Computer Science Department
Northwestern University and MIT

SIAM CSE 15
Salt Lake City
March 15, 2015

Motivation

New computer paradigm: large scale networks, local information collection and processing power

Argonne National Lab computer cluster

Parameter estimation in sensor networks

Multi-agent cooperative control and coordination

Smart grid state estimation systems

Distributed Multi-agent Optimization

- Connected undirected network: $\{1, \ldots, M\}$ nodes (agents, processors).
- Cooperatively solve

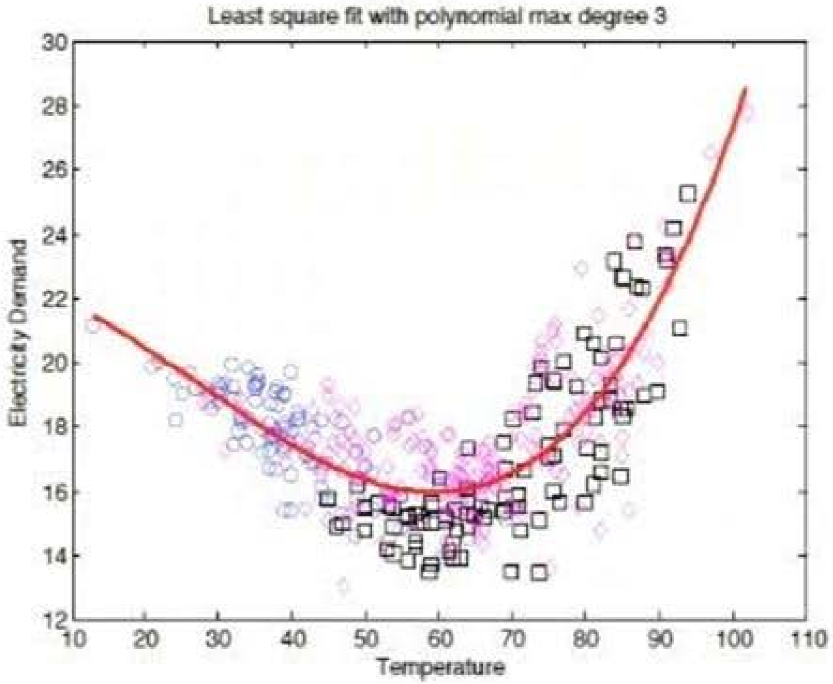
$$\begin{aligned} \min_{x} \quad & \sum_{i=1}^{M} f_i(x) \\ \text{s.t.} \quad & x \in \mathbb{R}^n. \end{aligned}$$

Machine Learning Example

- A network of 3 sensors, data collection: temperature t, electricity demand d.

- System goal: a 3rd-degree polynomial electricity demand model:

$$d(t) = x_3 t^3 + x_2 t^2 + x_1 t + x_0.$$



- System objective:

$$\min_{x} \sum_{i=1}^{3} ||A_i' x - d_i||_2^2.$$

where $A_i = [1, t_i, t_i^2, t_i^3]'$ at input data t_i.

Machine Learning General Set-up

- System objective: train weight vector x to

$$\min_{x} \quad \sum_{i=1}^{M-1} L_i(x) + p(x),$$

for some loss function L (on the prediction error) and penalty function p (on the complexity of the model).

  - Example: Least-Absolute Shrinkage and Selection Operator (LASSO):

$$\min_{x} \quad \sum_{i=1}^{M-1} ||A_i' x - b_i||_2^2 + \lambda \, ||x||_1 \, .$$

  - Other examples from ML estimation, low rank matrix completion, image recovery [Schizas, Ribeiro, Giannakis 08], [Recht, Fazel, Parrilo 10], [Steidl, Teuber, 10]

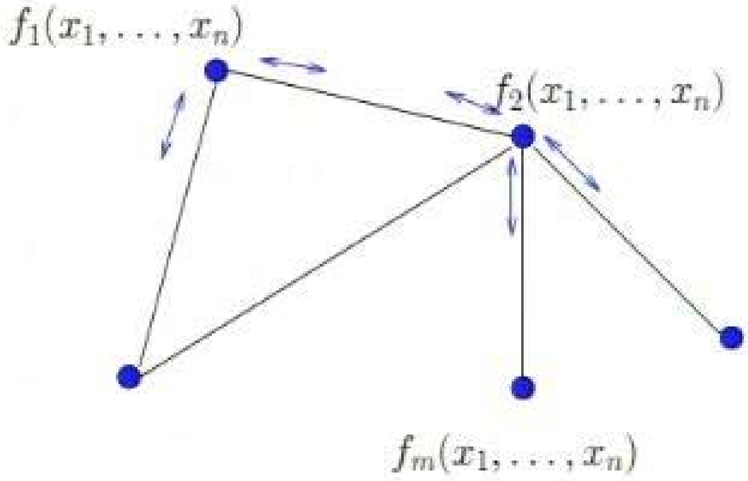
Distributed Multi-agent Optimization

- Connected undirected network: $\{1, \ldots, M\}$ nodes (agents).

- Cooperatively solve

$$\begin{aligned} \min_x \quad & \sum_{i=1}^{M} f_i(x) \\ \text{s.t.} \quad & x \in \mathbb{R}^n, \end{aligned}$$

  $f_i(x) : \mathbb{R}^n \to \mathbb{R}$ is a convex (possibly nonsmooth) function, known only to agent i.



- Distributed algorithm: each agent performing computations locally and communicating only to neighbors.

This Talk

- We present an asynchronous distributed ADMM-type algorithms for multi-agent optimization.

	Synchronous	Asynchronous
Centralized	O(1/k) [He, Yuan 11][1]	x
Distributed	$O(1/\sqrt{k})$	$O(1/\sqrt{k})$

[1] Under special assumptions (strong convexity, Lipschitz gradient), ADMM converges very fast [Goldfarb et al. 10], [Deng, Yin 12], [Hong, Luo 12].

Standard ADMM

- Standard ADMM solves a separable problem, where decision variable decomposes into two (linearly coupled) variables:

$$\begin{aligned} \min_{x,y} \quad & f(x) + g(y) \\ \text{s.t.} \quad & Ax + By = c. \end{aligned}$$

- Consider an Augmented Lagrangian function:

$$L_\beta(x, y, p) = f(x) + g(y) - p'(Ax + By - c) + \frac{\beta}{2} ||Ax + By - c||_2^2,$$

for some positive scalar β, dual variable p.

Standard ADMM

More specifically, updates are as follows:

$$
\begin{aligned}
x^{k+1} &= \text{argmin}_x \quad L_\beta(x, y^k, p^k), \\
y^{k+1} &= \text{argmin}_y \quad L_\beta(x^{k+1}, y, p^k), \\
p^{k+1} &= p^k - \beta(Ax^{k+1} - By^{k+1} - c).
\end{aligned}
$$

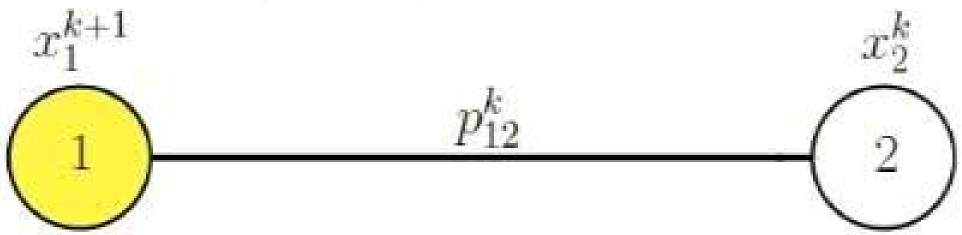
- Each minimization involves (quadratic perturbations of) functions f and g separately.
  - In many applications, these minimizations are easy (quadratic minimization, L_1 minimization, which arises in Huber fitting, basis pursuit, LASSO, total variation denoising). [Boyd et al. 10]

Special Case Study: 2-agent Optimization Problem

- Multi-agent optimization problem with two agents:

$$\begin{aligned} \min_{x_1,x_2} \quad & f_1(x_1) + f_2(x_2) \\ \text{s.t.} \quad & x_1 = x_2. \end{aligned}$$

- ADMM applied to this problem yields:



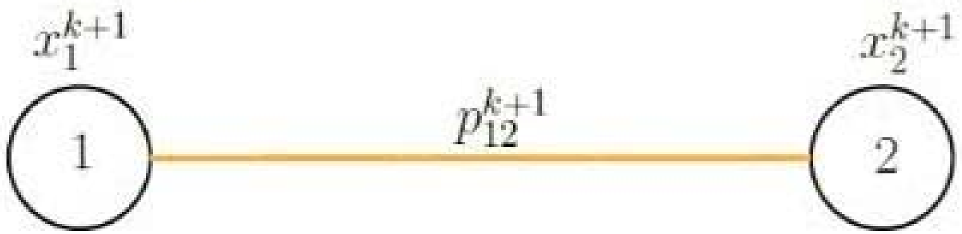
- $x_1^{k+1} = \text{argmin}_{x_1} \quad f_1(x_1) + f_2(x_2^k) - (p_{12}^k)'(x_1 - x_2^k) + \frac{\beta}{2} \left|\left|x_1 - x_2^k\right|\right|_2^2$

Special Case Study: 2-agent Optimization Problem

- Multi-agent optimization problem with two agents:

$$\begin{aligned} \min_{x_1, x_2} \quad & f_1(x_1) + f_2(x_2) \\ \text{s.t.} \quad & x_1 = x_2. \end{aligned}$$

- ADMM applied to this problem yields:

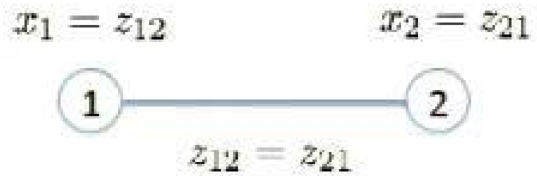


- $p_{12}^{k+1} = p_{12}^{k} - \beta(x_1^{k+1} - x_2^{k+1})$.

Multi-agent Optimization Problem: Reformulation

- Reformulate to remove ordering: technique from [Bertsekas, Tsitsiklis 89].
- Rewrite each constraint $x_i - x_j = 0$ for edge $e = (i, j)$ as

$$x_i = z_{ij}, \quad x_j = z_{ji},$$
$$z_{ij} = z_{ji}.$$



- Augmented Lagrangian

$$L_\beta(x, z, p) = f_1(x_1) + f_2(x_2) + p_{12}(x_1 - z_{12}) + p_{21}(x_2 - z_{21})$$
$$+ \frac{\beta}{2}\left((x_1 - z_{12})^2 + (x_2 - z_{21})^2\right).$$

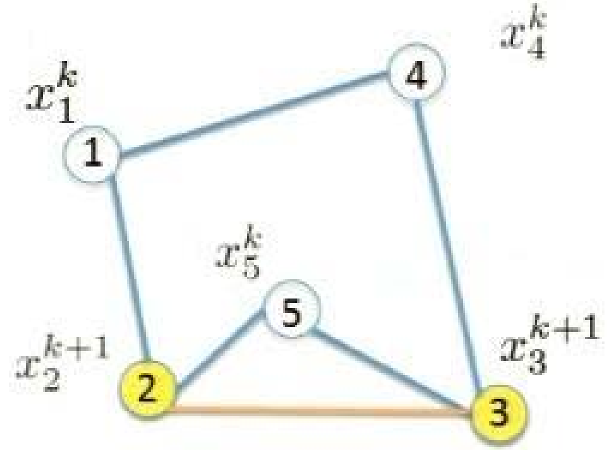
- Constraint set Z: $\{z_{ij} = z_{ji}\}$.

Multi-agent Asynchronous ADMM - Illustration

$$\min_{x,z} \quad \sum_{i=1}^{M} f_i(x_i)$$

$$\text{s.t.} \quad x_i = z_{ij}, \quad x_j = z_{ji} \quad \text{for } (i,j) \in E,$$

$$z \in Z.$$



- Set $Z = \{z \mid z_{ij} = z_{ji} \text{ for all } e = (i,j)\}$.
- We associate a Poisson local clock with each edge.
- If the clock corresponding to edge (i,j) ticks: increase iteration count by 1, the agents i, j and edge (i,j) become active.
- No adjustments for activation frequencies necessary.

Asynchronous ADMM Algorithm

At time step k, an edge $e = (i, j)$ and its end points become active.

a For $q = i, j$, the active primal variable x_q is updated as

$$x_q^{k+1} \in \operatorname*{argmin}_{x_q} f_q(x_q) - \sum_{t \in \mathcal{N}(q)} (p_{qt}^k)' x_q + \frac{\beta}{2} \left\| \sum_{t \in \mathcal{N}(q)} x_q - z_{qt}^k \right\|^2 .$$

The value x_q^{k+1} is sent to all active neighbors.

b Active agents i and j compute z_{ij}^{k+1} and z_{ji}^{k+1} as the components of z^{k+1}

$$z^{k+1} \in \operatorname*{argmin}_{z \in Z} -(p^k)'z + \frac{\beta}{2} \left\| \sum_{i,j} x_i^{k+1} - z_{ij} \right\|^2 .$$

c Agent i and j individually update active dual variable as

$$p_{ij}^{k+1} = p_{ij}^k + \frac{\beta}{2}(x_i^{k+1} - x_j^{k+1});$$

$$p_{ji}^{k+1} = p_{ji}^k + \frac{\beta}{2}(x_j^{k+1} - x_i^{k+1}).$$

Variables z_{qt} and p_{qt} are used whenever node q is active, but only updated when both q and t are active.

Convergence

Theorem

Under the assumption that each edge updates infinitely often, the iterates generated by the asynchronous ADMM algorithm converge to a primal-dual optimal solution almost surely.

Theorem

The following hold at each iteration k:

$$||\mathbb{E}(F(\bar{x}^k)) - F(x^*)|| \leq \frac{\alpha}{k},$$

where α is a constant and $\bar{x}^k = \frac{\sum_{l=0}^{k-1} x^l}{k}$, $\quad \bar{z}^k = \frac{\sum_{l=0}^{k-1} z^l}{k}$ are the ergotic average.

A similar rate result holds for the constraint violation $||\mathbb{E}(D\bar{x}(k) + H\bar{z}(k))||$.

Dependence of Convergence Rate on Network

Theorem

With initialization $x^0 = 0$, $z^0 = 0$ and $p^0 = 0$ and all edges active at every iterate, we have

$$|F(\bar{x}^k) - F(x^*)| \leq \frac{N\beta \|x^*\|^2}{Mk} + \frac{4C^2}{k\beta\rho_2(L(G))}.$$

$$\|D\bar{x}(k) - \bar{z}(k)\| \leq \frac{1}{2k\beta}\left(\frac{C\sqrt{2}}{\sqrt{\rho_2(L(G))}} + 1\right)^2 + \frac{N\beta \|x^*\|^2}{Mk}.$$

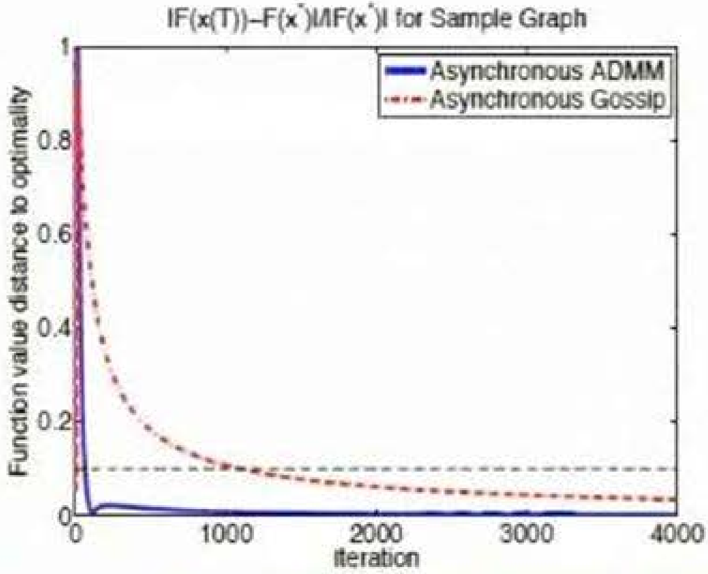
where N is the number of edges, $Dx - z = 0$ is the equality constraints, $\rho_2(L(G))$ is the second smallest positive eigenvalue of the Laplacian matrix $L(G)$ of the underlying graph. [2]

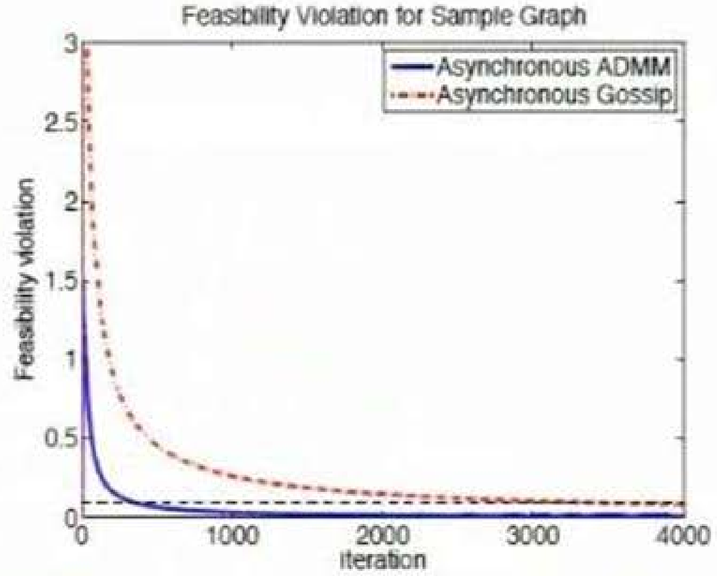
- Dependence on the algebraic connectivity of the graph: the more connected it is, the better the performance.

---

[2] $L(G)$ is a matrix with elements $[L(G)]_{ij} = \begin{cases} \text{degree}(i) & i = j, \\ -1 & (i,j) \in E. \end{cases}$

Simulations





	Number of Iterations to 10% of Optimal Function Value	Number of Iterations to Feasibility Violation < 0.1
Asynchronous ADMM	65	336
Asynchronous Gossip	1100	3252

Conclusions and Future Work

- For general convex problems, we developed an asynchronous distributed ADMM algorithms, which converges at the best known rate $O(1/k)$.

	Synchronous	Asynchronous
Centralized	O(1/k)	x
Distributed	O(1/k)	O(1/k)

- Simulation results illustrate the superior performance of ADMM (even for network topologies with slow mixing).
- Ongoing and Future Work:
  - Analyze graph topology effects on asynchronous ADMM algorithm performance.
  - Extension to directed graph, communication and computation noise.
  - ADMM type algorithm for time-varying graph topology.
  - Effect of global constraints.