First-order methods for distributed in-network optimization

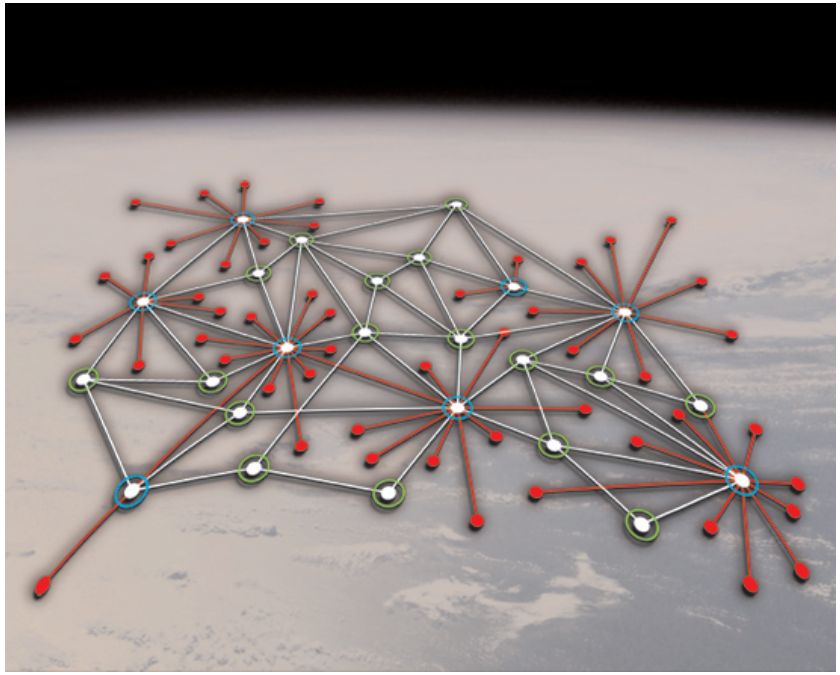
Angelia Nedić

angelia@illinois.edu

Industrial and Enterprise Systems Engineering Department
and Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

Large Networked Systems

- The recent advances in wired and wireless technology lead to the emergence of large-scale networks
  - Internet
  - Mobile ad-hoc networks
  - Wireless sensor networks
- The advances gave rise to new network applications including
  - Decentralized network operations including resource allocation, coordination, learning, estimation
  - Data-base networks
  - Social and economic networks



- As a result, there is a necessity to develop new models and tools for the design and performance analysis of such large complex dynamics systems

Abstract Computational Model

- Networked system thought of a collection of nodes or agents which can be sensors, computers, etc.

- Each agent has some capabilities to
  - collect and store the information,
  - process the information, and
  - locally communicate with some of the other agents

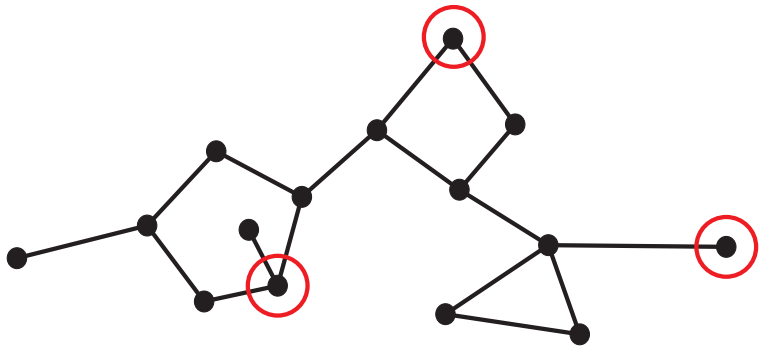
- At any instant of time, we represent the system with a time-varying graph where the edge set captures neighbor relations
  - the edges can be directed or undirected

- The networked system can be though of as a computational system where a global network-wide task is to be performed while using the agents/nodes resources and the local communications to achieve the global task

global task modeled as a network optimization problem

- MAIN ISSUE: the locality of the information - absence of central access to all the information
- MAIN IDEA: use local information exchange to spread the information through the entire network
- PRICE/BOTTLENECK: the agility of the system is heavily affected by the network:
  - the network ability to spread the information (connectivity topology)
  - the network reliability - communication medium (noisy links, links prone to failure)
  - the communication protocol that the network is using

Example 1: Computing Aggregates in P2P Networks



- Distributedly compute the average size of the files stored?*

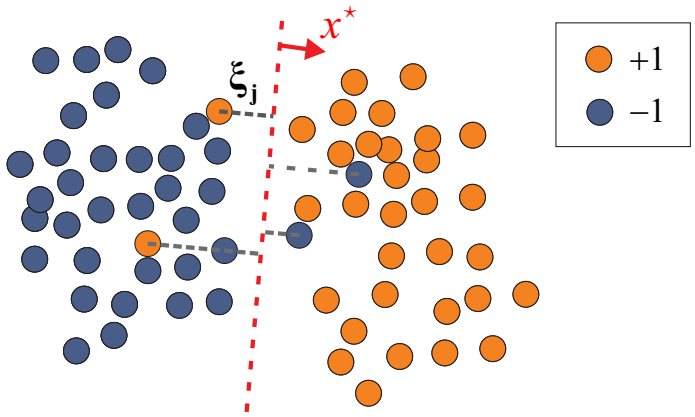
$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{m} \|x-\theta_i\|^2$$

  - θ_i is the average size of the files at location i
  - The value θ_i is known at that location only
  - No central access to all θ_i, $i = 1, \ldots, m$.

*D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in Proc. of 44th Annual IEEE Symposium on Foundations of CS, pp. 482-491, Oct. 2003.

Example 2: Support Vector Machine (SVM) - Centralized Case

Given a data set $\{a_j, b_j\}_{j=1}^n$, where $a_j \in \mathbb{R}^d$ and $b_j \in \{+1, -1\}$



- Find a maximum margin separating hyperplane $x^\star$

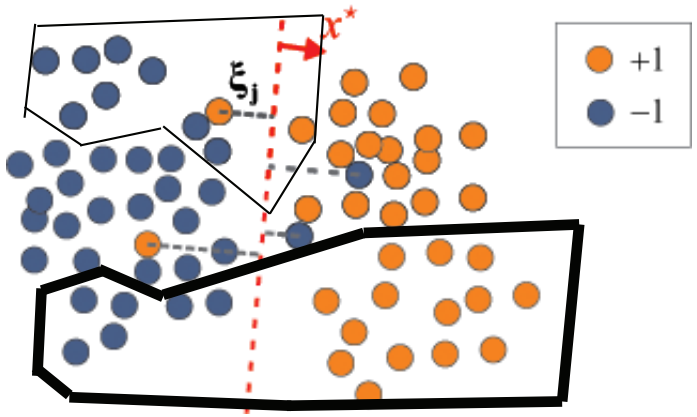
Centralized (not distributed) formulation

$$\min_{x\in\mathbb{R}^d, \xi\in\mathbb{R}^n} f(x,\xi) \triangleq \frac{1}{2}\|x\|^2 + C\sum_{j=1}^n \xi_j$$

$$\text{s.t. } (x,\xi) \in X \triangleq \{(x,\xi) \mid b_j\langle x, a_j\rangle \geq 1 - \xi_j, \xi_j \geq 0, \ \forall j = 1,\ldots,n\}$$

Example 2: Support Vector Machine (SVM) - Decentralized Case

Given m locations, each location i with its data set $\{a_j, b_j\}_{j \in J_i}$, where $a_j \in \mathbb{R}^d$ and $b_j \in \{+1, -1\}$



- Find a maximum margin separating hyperplane $x^\star$, without disclosing the data sets

$$\min_{x \in \mathbb{R}^d, \xi \in \mathbb{R}^n} \sum_{i=1}^{m} \left(\frac{1}{2m} \|x\|^2 + C \sum_{j \in J_i} \xi_j \right)$$

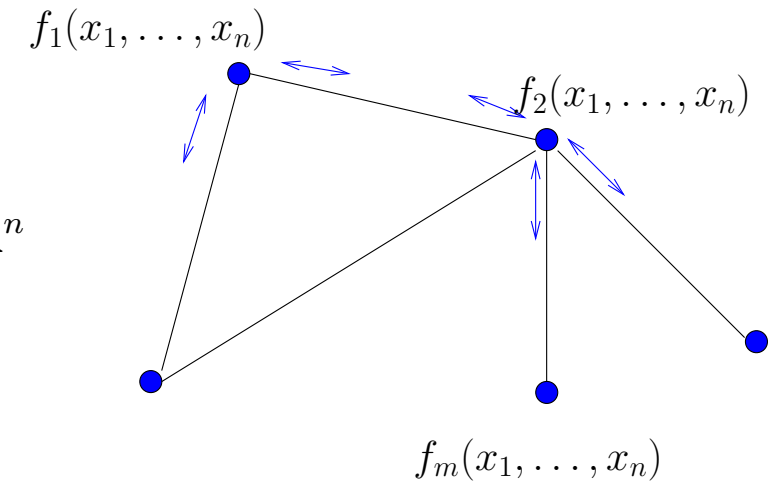
$$\text{s.t. } (x, \xi) \in \cap_{i=1}^{m} X_i,$$

$$X_i \triangleq \{(x, \xi) \mid b_j \langle x, a_j \rangle \geq 1 - \xi_j, \xi_j \geq 0, \ \forall j \in J_i\}$$

General Model

- Network of m agents represented by an undirected graph $([m], \mathscr{E})$ where $[m] = \{1, \ldots, m\}$ and $\mathscr{E}$ is the edge set
- Each agent i has an objective function $f_i(x)$ known to that agent only
- Common constraint (closed convex) set X known to all agents

Distributed Self-organized Agent System



The problem can be formalized:

minimize $\sum_{i=1}^{m} f_i(x)$ subject to $x \in X \subseteq \mathbb{R}^n$

How Agents Manage to Optimize Global Network Problem?

$$\text{minimize} \sum_{i=1}^{m} f_i(x) \quad \text{subject to } x \in X \subseteq \mathbb{R}^n$$

- Each agent i will generate its own estimate $x_i(t)$ of an optimal solution to the problem

- Each agent will update its estimate $x_i(t)$ by performing two steps:
  - Consensus-like step (mechanism to align agents estimates toward a common point)
  - Local gradient-based step (to minimize its own objective function)

C. Lopes and A. H. Sayed, "Distributed processing over adaptive networks," Proc. Adaptive Sensor Array Processing Workshop, MIT Lincoln Laboratory, MA, June 2006.

A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E90-A, no. 8, pp. 1504-1510, 2007.

A. Nedić and A. Ozdaglar "On the Rate of Convergence of Distributed Asynchronous Subgradient Methods for Multi-agent Optimization" Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, USA, 2007, pp. 4711-4716.

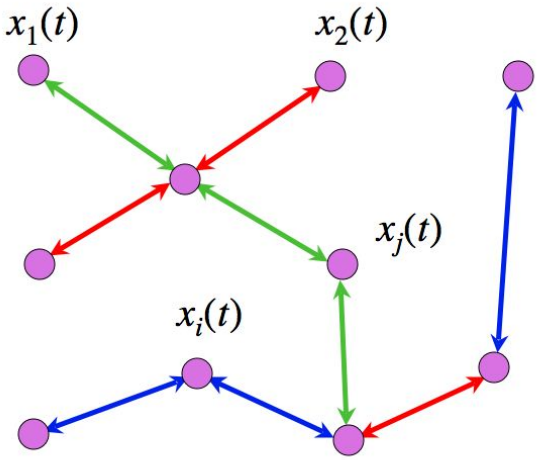
A. Nedić and A. Ozdaglar, Distributed Subgradient Methods for Multi-agent Optimization IEEE Transactions on Automatic Control 54 (1) 48-61, 2009.

Consensus Problem

Consider a connected network of m-agent, each knowing its own scalar value $x_i(0)$ at time $t = 0$.

The problem is to design a distributed and local algorithm ensuring that **the agents agree on the same value** x, i.e.,

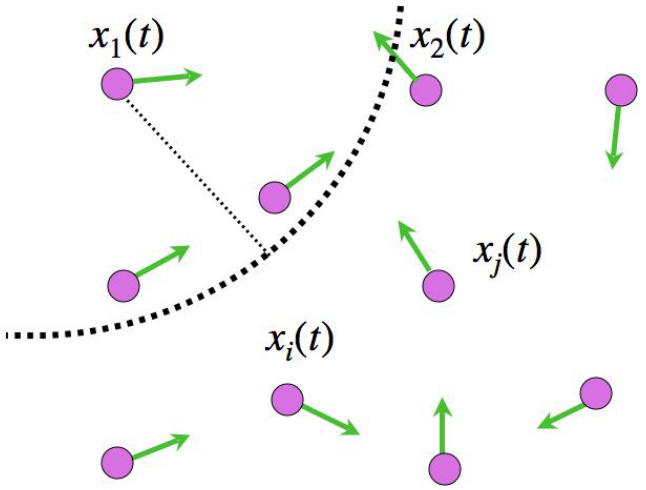
$$\lim_{t\to\infty} x_i(t) = x \qquad \text{for all } i.$$



Leaderless Heading Alignment

A system of autonomous agents are moving in the plane with the same speed but with different headings [Vicsek 95, Jadbabaie *et al.* 03]

The objective is to design a local protocol that will ensure the alignment of agent headings



Consensus Algorithm

Each agent combines its estimate $x_i(t)$ with the estimates $x_j(t)$ received from its neighbors

$$x_i(t+1) = \sum_{j \in N_i} a_{ij}\, x_j(t) \qquad \text{for all } i.$$

where N_i is the set of neighbors of agent i (including itself)

$$N_i = \{j \in [m] \mid (i,j) \in \mathscr{E}\}$$

$a_{ij} \geq 0$ is a weight that agent i assigns to the estimate coming from its neighbor $j \in N_i$.

The weights $\{a_{ij}, j \in N_i\}$ sum to 1, i.e., $\sum_{j \in N_i} a_{ij} = 1$ for all agents i

Introducing the values $a_{ij} = 0$ when $j \notin N_i$, the consensus algorithm can be written as:

$$x_i(t+1) = \sum_{j=1}^{m} a_{ij}\, x_j(t)$$

where

$$a_{ij} \geq 0 \qquad \text{with } a_{ij} = 0 \text{ when } j \notin N_i$$

$$\sum_{j=1}^{m} a_{ij} = 1$$

Under suitable conditions, the iterate sequences $\{x_i(t)\}$ converge to the same limit point.

Distributed Optimization Algorithm

$$\text{minimize } \sum_{i=1}^{m} f_i(x) \text{ subject to } x \in X \subseteq \mathbb{R}^n$$

- At time t, each agent i has its own estimate $x_i(t)$ of an optimal solution to the problem

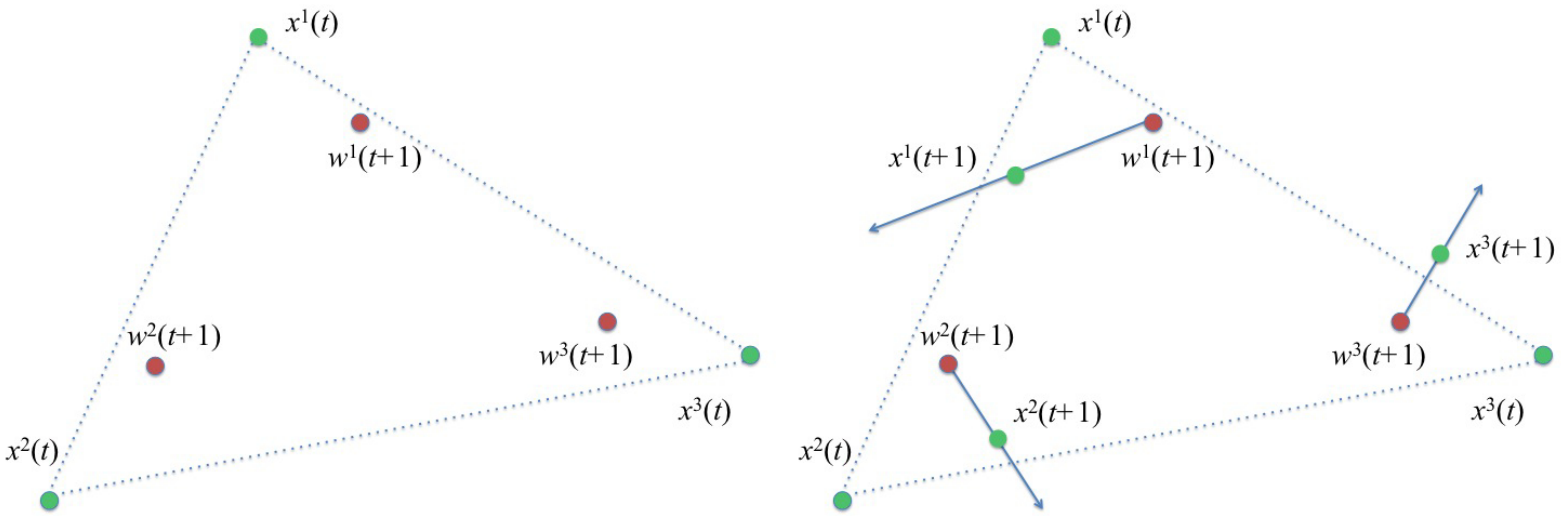
- At time $t+1$, agents communicate their estimates to their neighbors and update by performing two steps:
  - **Consensus-like step** to mix their own estimate with those received from neighbors

$$w_i(t+1) = \sum_{j=1}^{m} a_{ij} x_j(t) \qquad (a_{ij} = 0 \text{ when } j \notin N_i)$$

  - Followed by **a local gradient-based step**

$$x_i(t+1) = \Pi_X[w_i(t+1) - \alpha(t)\nabla f_i(w_i(t+1))]$$

where $\Pi_X[y]$ is the Euclidean projection of y on X, f_i is the local objective of agent i and $\alpha(t) > 0$ is a stepsize



Intuition Behind the Algorithm: It can be viewed as a consensus steered by a "force":

$$
\begin{aligned}
x_i(t+1) &= w_i(t+1) + (\Pi_X[w_i(t+1) - \alpha(t)\nabla f_i(w_i(t+1))] - w_i(t+1)) \\
&= w_i(t+1) + \underbrace{(\Pi_X[w_i(t+1) - \alpha(t)\nabla f_i(w_i(t+1))] - \Pi_X[w_i(t+1)])}_{\text{small stepsize } \alpha(t)} \\
&\approx w_i(t+1) - \alpha(t)\nabla f_i(w_i(t+1)) \\
&= \sum_{j=1}^{m} a_{ij}x_j(t) - \alpha(t)\nabla f_i\left(\sum_{j=1}^{m} a_{ij}x_j(t)\right)
\end{aligned}
$$

Matrices A that lead to consensus, also yield convergence of an optimization algorithm

Convergence Result for Static Network

Convex Problem: Let X be closed and convex, and each $f_i : \mathbb{R}^n \to \mathbb{R}$ be convex with bounded (sub)gradients over X. Assume the problem $\min_{x \in X} \sum_{i=1}^m f_i(x)$ has a solution.

Stepsize Rule: Let the stepsize $\alpha(t)$ be such that $\sum_{t=0}^{\infty} \alpha(t) = \infty$ and $\sum_{t=0}^{\infty} \alpha^2(t) < \infty$.

Network: Let the graph $([m], \mathscr{E})$ be directed and strongly connected. Let the matrix $A = [a_{ij}]$ of agents' weights be **doubly stochastic**. Then,

$$\lim_{t \to \infty} x_i(t) = x^* \quad \text{for all } i,$$

where x^* is a solution of the problem.

Proof Outline:

Use $\sum_{i=1}^m \|x_i(t) - x^*\|^2$ as a Lyapunov function, where x^* is a solution to the problem

Due to convexity and (sub)gradient boundedness, we have

$$\sum_{i=1}^m \|x_i(t+1)-x^*\|^2 \leq \sum_{i=1}^m \|w_i(t+1)-x^*\|^2-2\alpha(t)\sum_{i=1}^m (f_i(w_i(t+1)) - f_i(x^*))+\alpha^2(t)C^2$$

By $w_i(t+1) = \sum_{j=1}^m a_{ij}\, x_j(t)$ and the doubly stochasticity of A, we have

$$\sum_{i=1}^m \|x_i(t+1)-x^*\|^2 \leq \sum_{j=1}^m \|x_j(t)-x^*\|^2-2\alpha(t)\sum_{i=1}^m (f_i(w_i(t+1)) - f_i(x^*))+\alpha^2(t)C^2$$

Thus, letting $s(t+1) = \frac{1}{m}\sum_{i=1}^m x_i(t+1)$ we see

$$\sum_{i=1}^m \|x_i(t+1) - x^*\|^2 \leq \sum_{j=1}^m \|x_j(t) - x^*\|^2 - 2\alpha(t)\sum_{i=1}^m (f_i(s(t+1)) - f_i(x^*))$$
$$+ 2\alpha(t)\sum_{i=1}^m (f_i(s(t+1)) - f_i(w_i(t+1))) + \alpha^2(t)C^2$$

Letting $F(x) = \sum_{i=1}^m f_i(x)$ and using (sub)gradient boundedness, we find

$$\underbrace{\sum_{i=1}^m \|x_i(t+1) - x^*\|^2}_{V(t+1)} \leq \underbrace{\sum_{j=1}^m \|x_j(t) - x^*\|^2}_{V(t)} - 2\alpha(t) \underbrace{(F(s(t+1)) - F(x^*))}_{\geq 0}$$
$$+ 2\alpha(t)C \sum_{i=1}^m \|s(t+1) - w_i(t+1)\| + \alpha^2(t)C^2$$

We can see $\sum_{t=0}^\infty \alpha(t)C \sum_{i=1}^m \|s(t+1) - w_i(t+1)\| < \infty$

The result would hold if we can show $\|s(t+1) - w_i(t+1)\| \to 0$ as $t \to \infty$ for all i

The trouble is in showing $\|s(t+1) - w_i(t+1)\| \to 0$ as $t \to \infty$ for all i, which is exactly where the **network impact is** – it is important to know that the network is diffusing (mixing) the information fast enough.

Formally speaking, the rate of convergence of A^t to its limit is critical.

When the network is connected, the matrices A^t converge to the matrix $\frac{1}{m}\mathbf{1}\mathbf{1}'$, as $t \to \infty$

The convergence rate is

$$\left| [A^t]_{ij} - \frac{1}{m} \right| \leq q^t, \qquad \text{where } q \in (0,1)$$

We have for arbitrary $0 \leq \tau < t$

$$x_i(t+1) = w^i(t+1) + (\underbrace{\Pi_X[w_i(t+1) - \alpha(t)\nabla f_i(w_i(t+1))] - w_i(t+1)}_{e_i(t)})$$

$$= \sum_{j=1}^{m} a_{ij}\, x_j(t) + e_i(t) = \cdots$$

$$= \sum_{j=1}^{m} [A^{t+1-\tau}]_{ij}\, x_j(\tau) + \sum_{k=\tau+1}^{t} \sum_{j=1}^{m} [A^k]_{ij} e_j(t-k) + e_i(t)$$

Similarly, for $s(t+1) = \frac{1}{m}\sum_{i=1}^{m} x_i(t+1)$ we have

$$s(t+1) = s(t) + \frac{1}{m}\sum_{j=1}^{m} e_j(t) = \cdots = \sum_{j=1}^{m} \frac{1}{m} x_j(\tau) + \sum_{k=\tau+1}^{t} \sum_{j=1}^{m} \frac{1}{m} e_j(t-k) + \sum_{j=1}^{m} \frac{1}{m} e_j(t)$$

Thus,

$$\|x_i(t+1) - s(t+1)\| \leq q^{t+1-\tau} \sum_{j=1}^{m} \|x_j(\tau)\| + \sum_{k=\tau+1}^{t} \sum_{j=1}^{m} q^k \|e_j(t-k)\| + \sum_{j=1}^{m} \frac{1}{m} \|e_j(t)\| + \|e_i(t)\|$$

Since

$$e_i(t) = \Pi_X[w_i(t+1) - \alpha(t)\nabla f_i(w_i(t+1))] - w_i(t+1)$$

we have

$$\|e_i(t)\| \leq \alpha(t)C$$

Hence

$$\|x_i(t+1) - s(t+1)\| \leq q^{t+1-\tau} \sum_{j=1}^{m} \|x_j(\tau)\| + mC \sum_{k=\tau+1}^{t} q^k \alpha(t-k) + 2C\alpha(t)$$

By choosing τ such that $\|e(t)\| \leq \epsilon$ for all $t \geq \tau$ and then, using some properties of the sequences involved in the above relation, we show

$$\|x_i(t+1) - s(t+1)\| \to 0$$

which in view of $x_i(t+1) = w_i(t+1) + e_i(t)$ and $e_i(t) \to 0$ implies

$$\|w_i(t+1) - s(t+1)\| \to 0$$

Convergence Result for Time-varying Networks

- Consensus-like step to mix their own estimate with those received from neighbors

$$w_i(t+1) = \sum_{j=1}^{m} a_{ij}(t) x_j(t) \qquad (a_{ij}(t) = 0 \text{ when } j \notin N_i(t))$$

- Followed by a local gradient-projection step

$$x_i(t+1) = \Pi_X[w_i(t+1) - \alpha(t)\nabla f_i(w_i(t+1))]$$

For convergence, some conditions on the weight matrices $A(t) = [a_{ij}(t)]$ are needed. *Convergence Result for Time-varying Network* Let the problem be convex, f_i have bounded (sub)gradients on X, and $\sum_{t=0}^{\infty} \alpha(t) = \infty$ and $\sum_{t=0}^{\infty} \alpha^2(t) < \infty$. Let the graphs $G(t) = ([m], \mathscr{E}(t))$ be directed and strongly connected, and the matrices $A(t)$ be such that $a_{ij}(t) = 0$ if $j \notin N_i(t)$, while $a_{ij}(t) \geq \gamma$ whenever $a_{ij}(t) > 0$, where $\gamma > 0$. Also assume that $A(t)$ are doubly stochastic[†]. Then,

$$\lim_{t\to\infty} x_i(t) = x^* \quad \text{for all } i,$$

where x^* is a solution of the problem.

[†]J. N. Tsitsiklis, "Problems in Decentralized Decision Making and Computation," Ph.D. Thesis, Department of EECS, MIT, November 1984; technical report LIDS-TH-1424, Laboratory for Information and Decision Systems, MIT

Related Papers

- AN and A. Ozdaglar "Distributed Subgradient Methods for Multi-agent Optimization" *IEEE Transactions on Automatic Control* 54 (1) 48-61, 2009.
  The paper looks at a basic (sub)gradient method with a constant stepsize
- S.S. Ram, AN, and V.V. Veeravalli "Distributed Stochastic Subgradient Projection Algorithms for Convex Optimization." *Journal of Optimization Theory and Applications* 147 (3) 516-545, 2010.
  The paper looks at stochastic (sub)gradient method with diminishing stepsizes and constant as well
- S.S. Ram, A. Nedic, and V.V. Veeravalli "A New Class of Distributed Optimization Algorithms: Application to Regression of Distributed Data," *Optimization Methods and Software* 27(1) 71–88, 2012.
  The paper looks at extension of the method for other types of network objective functions

Other Extensions

$$w_i(t+1) = \sum_{j=1}^{m} a_{ij}(t) x_j(t) \qquad (a_{ij}(t) = 0 \text{ when } j \notin N_i(t))$$

$$x_i(t+1) = \Pi_X[w_i(t+1) - \alpha(t)\nabla f_i(w_i(t+1))]$$

Extensions include

- Gradient directions $\nabla f_i(w_i(t+1))$ can be erroneous

  $$x_i(t+1) = \Pi_X[w_i(t+1) - \alpha(t)(\nabla f_i(w_i(t+1) + \varphi_i(t+1))]$$

  [Ram, Nedić, Veeravali 2009, 2010, Srivastava and Nedić 2011]

- The links can be noisy i.e., $x_j(t)$ is sent to agent i, but the agent receives $x_j(t) + \epsilon_{ij}(t)$ [Srivastava and Nedić 2011]

- The updates can be asynchronous; the edge set $\mathscr{E}(t)$ is random [Ram, Nedić, and Veeravalli - gossip, Nedić 2011]

- The set X can be $X = \cap_{i=1}^{m} X_i$ where each X_i is a private information of agent i

$$x_i(t+1) = \Pi_{X_i}[w_i(t+1) - \alpha(t)\nabla f_i(w_i(t+1))]$$

[Nedić, Ozdaglar, and Parrilo 2010, Srivastava[‡] and Nedić 2011, Lee and AN 2013]

- Different sum-based functional structures [Ram, Nedić, and Veeravalli 2012]

S. S. Ram, AN, and V.V. Veeravalli, "Asynchronous Gossip Algorithms for Stochastic Optimization: Constant Stepsize Analysis," in Recent Advances in Optimization and its Applications in Engineering, the 14th Belgian-French-German Conference on Optimization (BFG), M. Diehl, F. Glineur, E. Jarlebring and W. Michiels (Eds.), 2010, pp. 51-60.

A. Nedić "Asynchronous Broadcast-Based Convex Optimization over a Network," *IEEE Transactions on Automatic Control* 56 (6) 1337-1351, 2011.

S. Lee and A. Nedić "Distributed Random Projection Algorithm for Convex Optimization," *IEEE Journal of Selected Topics in Signal Processing*, a special issue on Adaptation and Learning over Complex Networks, 7, 221-229, 2013

K. Srivastava and A. Nedić "Distributed Asynchronous Constrained Stochastic Optimization," *IEEE Journal of Selected Topics in Signal Processing* 5 (4) 772-790, 2011.

[‡]Uses different weights

RETURN TO: Support Vector Machine (SVM)

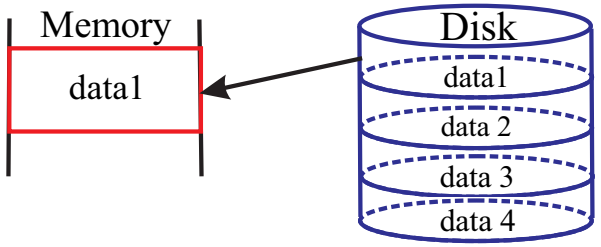
- **Challenge 1**: Online Learning (the constraint set X not known in advance)
  - Standard gradient projection cannot be used

$$x(k+1) = \Pi_X[x(k) - \alpha_k \nabla f(x(k))],$$

  where $\Pi_X[x]$ is the projection of x on the set X.
- **Challenge 2**: Large number of components
  - Even approximation (alternate projections) is intractable

$$\Pi_X[\cdot] \approx \Pi_{X_n}[\cdots \Pi_{X_2}[\Pi_{X_1}[\cdot]]]$$

  where $X_j = \{(x, \xi) \mid b_j \langle x, a_j \rangle \geq 1 - \xi_j, \xi_j \geq 0\}$
- **Challenge 3**: What if data cannot fit in memory?
  - Multiple disk I/Os



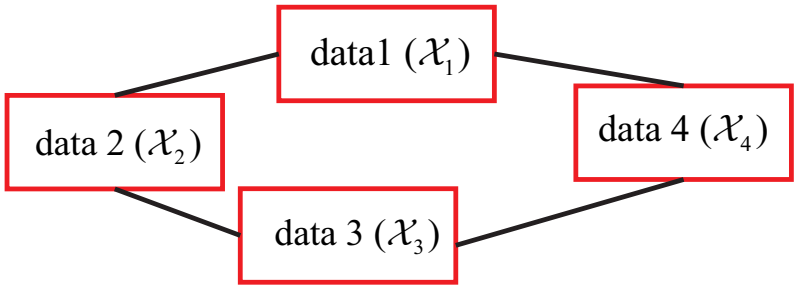
- Distributed formulation

$$\text{minimize} \sum_{i=1}^{m} f_i(x,\xi), \quad f_i(x,\xi) = \frac{1}{2m}\|x\|^2 + C\sum_{j\in I_i}\xi_j$$

$$\text{subject to } (x,\xi) \in \cap_{i=1}^{m} X_i,$$

$$\text{where } X_i = \cap_{j\in I_i} X_i^j \quad \forall i = 1,\ldots,m$$

$$X_i^j = \{(x,\xi) \mid b_j\langle x, a_j\rangle \geq 1-\xi_j,\ \xi_j \geq 0\} \quad \forall j \in I_i$$

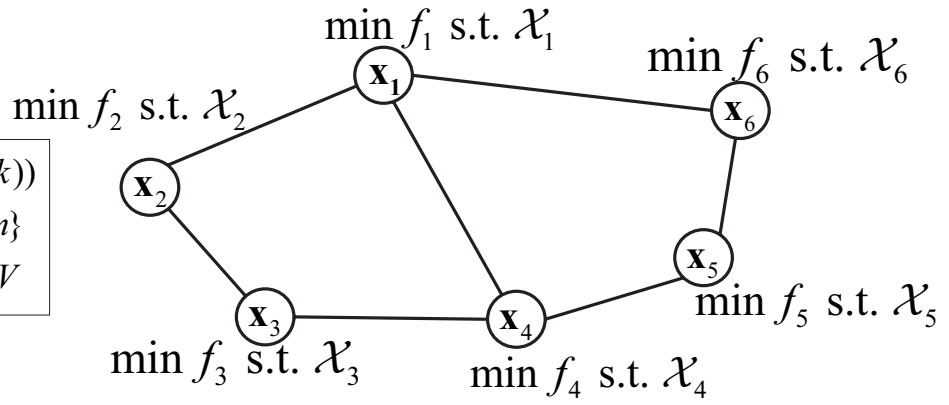


- **More efficient** if communication cost is low

Distributed Optimization in Network: Distributed Constraints

- Problem information distributed: each node (agent) knows f_i and X_i only
- Agents unaware of the network topology, only talk to immediate neighbors

$$G(k) = (V, E(k))$$
$$V = \{1, \ldots, m\}$$
$$E(k) \subseteq V \times V$$



- Goal: All agents to cooperatively solve
  - $\min f(x) = \sum_{i=1}^{m} f_i(x) \qquad \text{s.t. } x \in X \triangleq \bigcap_{i=1}^{m} X_i, \; X_i = \cap_{j \in I_i} X_i^j$

Previous Work on Distributed Optimization

- **Markov incremental algorithms** [§] [¶] [‖]
- **Distributed subgradient algorithms**[**] [††] [‡‡]
- None of them can handle on-line constraints
- All of them use an exact projection on X_i

[§]B. Johansson, M. Rabi, and M. Johansson, "A simple peer-to-peer algorithm for distributed optimization in sensor networks," in Proceedings of the 46th IEEE Conference on Decision and Control, Dec. 2007, pp. 4705–4710.

[¶]S. S. Ram, A. Nedić, and V. V. Veeravalli, "Incremental stochastic subgradient algorithms for convex optimization," SIAM J. on Optimization, vol. 20, no. 2, pp. 691–717, Jun. 2009.

[‖]J. Duchi, A. Agarwal, M. Johansson, M. Jordan, "Ergodic Mirror Descent," SIAM Journal on Optimization

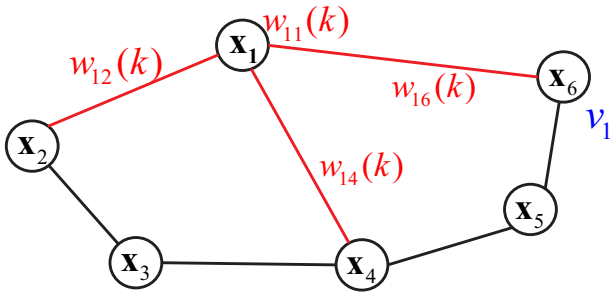
[**]K. Srivastava and A. Nedić, "Distributed asynchronous constrained stochastic optimization," IEEE J. Sel. Topics. Signal Process., vol. 5, no. 4, pp. 772–790, 2011.

[††]A. Nedić, A. Ozdaglar, and P. Parrilo, "Constrained consensus and optimization in multi-agent networks," IEEE Transactions on Automatic Control, vol. 55, no. 4, pp. 922–938, April 2010.

[‡‡]I. Lobel, A. Ozdaglar, and D. Feijer, "Distributed multi-agent optimization with state-dependent communication," Mathematical Programming, vol. 129, no. 2, pp. 255–284, 2011.

Distributed Random Projection (DRP) Algorithm

- Each agent i maintains an estimate sequence $\{x_i(k)\}$.



$$v_1(k) = w_{11}(k)x_1(k) + w_{12}(k)x_2(k) + w_{14}(k)x_4(k) + x_{16}(k)x_6(k)$$

- Initialize $x_i(0)$, for $i \in V$. For $k \geq 0$, each agent i does

1. **Mixing** $v_i(k) = \sum_{j=1}^{m} w_{ij}(k)x_j(k)$
2. **Gradient update** $\tilde{v}_i(k) = v_i(k) - \alpha_k \nabla f_i(v_i(k))$
3. **Projection** A random variable $\Omega_i(k) \in I_i$ is drawn, and a component $X_i^{\Omega_i(k)}$ of $X_i = \bigcap_{j \in I_i} X_i^j$ is used for projection.

$$x_i(k+1) = \Pi_{X_i^{\Omega_i(k)}}[\tilde{v}_i(k)]$$

Distributed Mini-Batch Random Projection (DMRP)

- What if $\mathcal{X}_\rangle$ consists of 10^4 hyperplanes?
  - 100 samples will provide a better approximation than a single sample
- Initialize $x_i(0)$, for $i \in V$. For $k \geq 0$, each agent i does the following

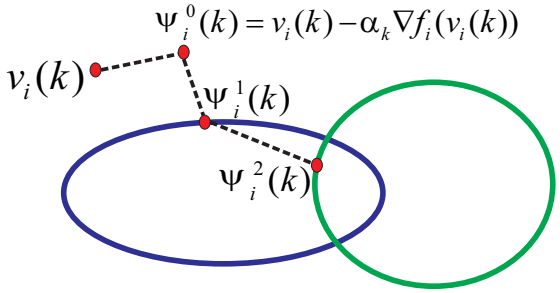
1. Mixing $v_i(k) = \sum_{j=1}^m w_{ij}(k) x_j(k)$

2. Gradient update $\psi_i^0(k) = v_i(k) - \alpha_k \nabla f_i(v_i(k))$

3. Projections A batch of b independent random variables $\Omega_i^r(k) \in I_i$, $r = 1, \ldots, b$ is drawn. The components $X_i^{\Omega_i^1(k)}, \ldots, X_i^{\Omega_i^b(k)}$ of $X_i = \bigcap_{j \in I_i} X_i^j$ are used for sequential projections.

$$\psi_i^r(k) = \Pi_{X_i^{\Omega_i^r(k)}}[\psi_i^{r-1}(k)], \quad \text{for } r = 1, \ldots, b$$

$$x_i(k+1) = \psi_i^b(k)$$



Almost Sure Convergence of DRP and DMRP

- Notations

$$f^* = \min_{x \in X} f(x), \qquad X^* = \{x \in X \mid f(x) = f^*\}$$

$$f(x) = \sum_{i=1}^{m} f_i(x), \qquad X = \cap_{i=1}^{m} X_i$$

Proposition 1

Let $\sum_{k=0}^{\infty} \alpha_k = \infty$ and $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$.

Assume that the problem has a nonempty optimal set X^*.

Then, with typical assumptions, the iterate sequences $\{x_i(k)\}$, $i = 1, \ldots, m$, generated by DRP (or DMRP) algorithm converge almost surely to some (common random) optimal point $x^\star \in X^*$, i.e.,

$$\lim_{k \to \infty} x_i(k) = x^\star \quad \text{for all } i = 1, \ldots, m \ a.s.$$

Assumptions on the Functions f_i and Sets X_i^j

Assumption 1

(a) The sets X_i^j, $j \in I_i$ are closed and convex for every i.

(b) Each function $f_i : \mathbb{R}^d \to \mathbb{R}$ is convex.

(c) The functions f_i, $i \in V$, are differentiable and have *Lipschitz gradients* with a constant L over $\mathbb{R}^d$,

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\| \quad \text{for all } x, y \in \mathbb{R}^d.$$

(d) The gradients $\nabla f_i(x)$ are bounded over the set X, i.e., there exists a constant G_f such that

$$\|\nabla f_i(x)\| \leq G_f \quad \text{for all } x \in X \text{ and all } i.$$

- Assumption 1(d) is satisfied, for example, when X is compact.

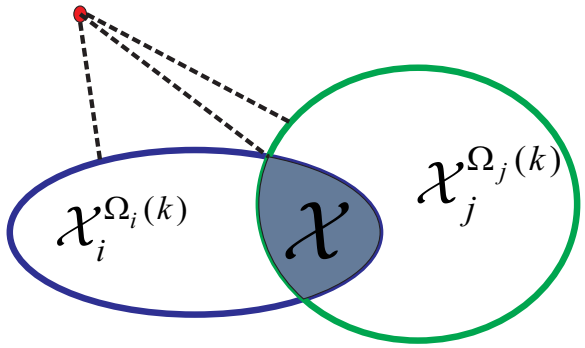
Assumption on the Set Process $\{\Omega_i(k)\}$

Assumption 2: Set Regularity

For all i, there exists a constant $c > 0$ such that

$$\mathrm{dist}^2(x, X) \leq c\mathsf{E}\left[\mathrm{dist}^2(x, X_i^{\Omega_i(k)})\right] \text{ for all } x \in \mathbb{R}^d.$$

- This is satisfied when
  - Each set X_i^j is given by a linear (in)equality.
  - X has a nonempty interior.



Assumptions on the Network $(V, E(k))$

Assumption 3 For all $k \geq 0$,

Network Connectivity

$\exists Q > 0$ such that the graph $\left(V, \bigcup_{\ell=0,\ldots,Q-1} E(k+\ell)\right)$ is strongly connected.

Doubly Stochasticity

(a) $[W(k)]_{ij} \geq 0$ and $[W(k)]_{ij} = 0$ when $j \notin N_i(k)$,

(b) $\sum_{j=1}^{m} [W(k)]_{ij} = 1$ for all $i \in V$,

(c) There exists a scalar $\eta \in (0, 1)$ such that $[W(k)]_{ij} \geq \eta$ when $j \in N_i(k)$,

(d) $\sum_{i=1}^{m} [W(k)]_{ij} = 1$ for all $j \in V$.

- Network is sufficiently connected.

- Each agent is equally influencing every other agent.

Simulation Results

DrSVM: D(M)RP applied on SVM

- Three text classification data sets

2*Data set	Statistics		
	n	d	s
astro-ph	62,369	99,757	0.08%
CCAT	804,414	47,236	0.16%
C11	804,414	47,236	0.16%

- Experimental set-up
  - 80% for training (equally divided among agents), 20% for testing
  - Stopping criteria
    - First run centralized random projection algorithm with $b = 1$
    - Set t_{acc} as the test accuracy of the final solution
    - Limit the total number of iterations
  - Stepsize: $\alpha_k = \frac{1}{k+1}$, Weights: $w_{ij}(k) = \frac{1}{|N_i(k)|}$

Simulation Results

- Table shows the number of iterations for all agents to reach the target accuracy t_{acc}
  - Graph topologies = clique, 3-regular expander graph
  - Batch sizes b = 1, 100, 1000
  - Number of agents m = 2, 6, 10
  - Maximum iteration = 20,000

Data set	t_{acc}	b	$m = 2$	Clique		3-regular expander	
				$m = 6$	$m = 10$	$m = 6$	$m = 10$
astro-ph	0.95	1	1,055	695	697	695	-
		100	11	8	11	11	11
		1000	2	2	2	2	2
CCAT	0.91	1	752	511	362	517	-
		100	11	10	8	10	8
		1000	2	3	2	3	3
C11	0.97	1	1,511	1,255	799	1,226	-
		100	16	17	12	17	15
		1000	2	2	2	2	2

When $m = 10$ each agent gets about 1,200 data points for $astro\text{-}ph$, and about 16,000 data points for $CCAT$ and $C11$

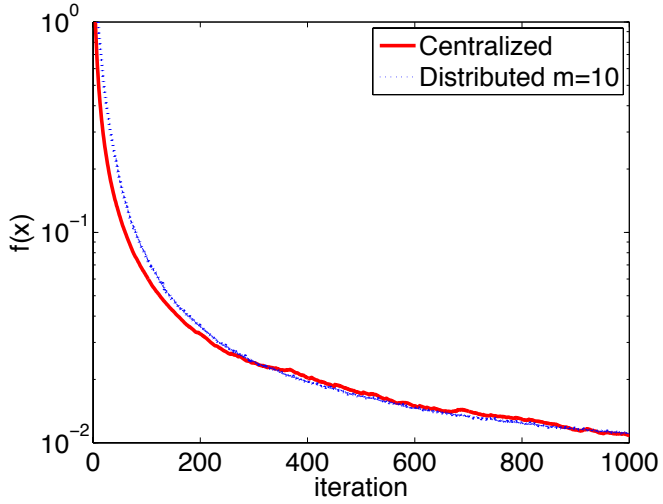
Simulation Results

- Repeated 100 times for the column: clique, m $= 10$

Data set	b	**Clique, $m = 10$**	$\bar{\mu}$	$\bar{\sigma}$	95% confidence
3*astro-ph	1	**697**	622.7	54.7	[611.8 633.5]
	100	**11**	9.9	1.2	[9.7 10.1]
	1000	**2**	2.1	0.2	[2.0 2.1]
3*CCAT	1	**362**	441.0	44.8	[432.1 449.9]
	100	**8**	8.2	0.9	[8.0 8.3]
	1000	**2**	2.5	0.5	[2.4 2.5]
3*C11	1	**799**	1126.4	181.1	[1090.5 1162.3]
	100	**12**	15.5	2.4	[15.1 16.0]
	1000	**2**	3.1	0.7	[3.0 3.3]

- The algorithm is more reliable for larger b

Simulation Results - `astro-ph`: Convergence to f^*



Compares $f(x)$ when $m = 1$ (centralized) and $m = 10$ with $b = 1$

Proof Sketch: Part I

Lemma 1: Projection Error

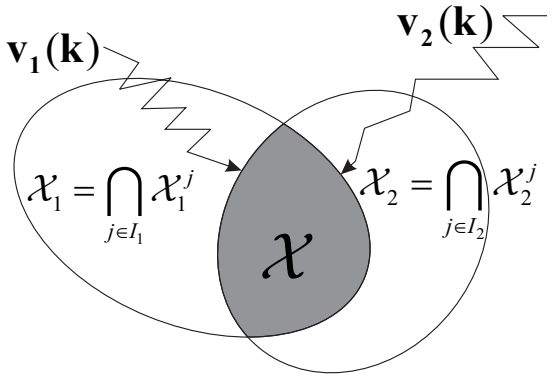
Let Assumption 1-3 hold. Let $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$.

Then,

$$\sum_{k=0}^{\infty} \text{dist}^2(v_i(k), X) < \infty \qquad \text{for all } i \ a.s.$$

- Define $z_i(k) = \Pi_X[v_i(k)]$. This also means

$$\lim_{k\to\infty} \|v_i(k) - z_i(k)\| = 0 \qquad \text{for all } i \ a.s.$$



Proof Sketch: Part II

Lemma 2: Disagreement Estimate

Let Assumption 3 hold (network). Consider the iterates generated by

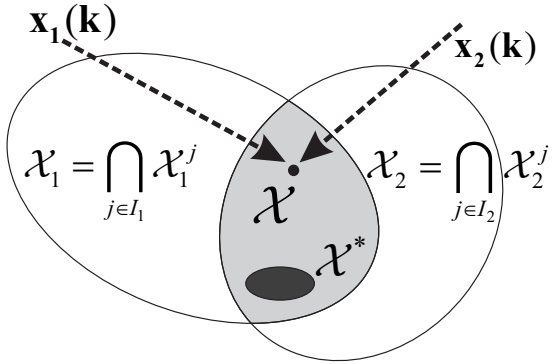
$$x_i(k+1) = \sum_{j=1}^{m} [W(k)]_{ij} x_j(k) + e_i(k) \quad \text{for all } i.$$

Suppose $\exists$ a sequence $\{\alpha_k\}$ such that $\sum_{k=0}^{\infty} \alpha_k \|e_i(k)\| < \infty$ for all i.

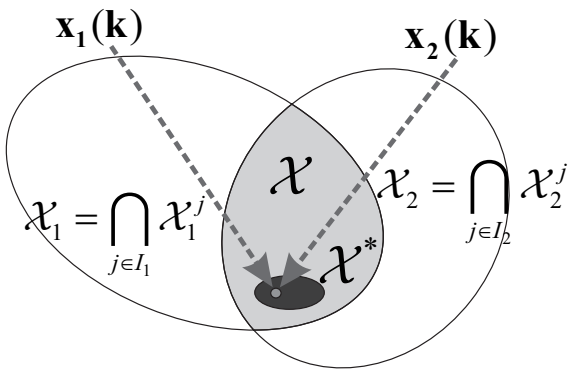
Then, for all i, j,

$$\sum_{k=0}^{\infty} \alpha_k \|x_i(k) - x_j(k)\| < \infty.$$

- Since $v_i(k) = \sum_{j=1}^{m} [W(k)]_{ij} x_j(k)$, we can define $e_i(k) = x_i(k+1) - v_i(k)$.
- $\sum_{k=0}^{\infty} \alpha_k \|e_i(k)\| < \infty$ from Lemma 1.



Proof Sketch: Part III



Convergence results (Robbins and Siegmund 1971)

Let $\{v_k\}$, $\{u_k\}$, $\{a_k\}$ and $\{b_k\}$ be sequences of non-negative random variables such that

$$\mathsf{E}[v_{k+1}|_k] \leq (1 + a_k)v_k - u_k + b_k \qquad \text{for all } k \geq 0 \quad a.s.,$$

where $_k$ denotes the collection $v_0, \ldots, v_k$, $u_0, \ldots, u_k$, $a_0, \ldots, a_k$ and $b_0, \ldots, b_k$.

Also, let $\sum_{k=0}^{\infty} a_k < \infty$ and $\sum_{k=0}^{\infty} b_k < \infty$ a.s.

Then, we have $\lim_{k\to\infty} v_k = v$ for a random variable $v \geq 0$ $a.s.$, and $\sum_{k=0}^{\infty} u_k < \infty$ $a.s.$

Proof Sketch: Part III

Basic Iterate Relation and Convergence

Let Assumption 1-3 hold. Let $\sum_{k=0}^{\infty} \alpha_k = \infty$ and $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$. For any $x^\star \in X^*$, we have

$$\mathsf{E}\left[\sum_{i=1}^{m} \|x_i(k+1) - x^\star\|^2 \mid \mathcal{F}_k\right] \leq (1 + A\alpha_k^2) \sum_{i=1}^{m} \|x_i(k) - x^\star\|^2 + mB\alpha_k^2 G_f^2$$
$$- 2\alpha_k (f(\bar{z}(k)) - f^*) + 4\alpha_k G_f \sum_{i=1}^{m} \|v_i(k) - \bar{v}(k)\|,$$

where $\bar{z}(k) = \frac{1}{m}\sum_{i=1}^{m} z_i(k)$ with $z_i(k) = \Pi_X[v_i(k)]$. Hence, $\left\{\sum_{i=1}^{m} \|x_i(k) - x^\star\|^2\right\}$ is convergent for every $x^\star \in X^*$, and

$$\sum_{k=0}^{\infty} \alpha_k (f(\bar{z}(k)) - f^*) < \infty.$$

From Lemma 1-2 and the continuity of f, we have

$$\lim_{k\to\infty} x_i(k) = x^\star \quad \text{for all } i \ a.s.$$

Removal of Doubly Stochastic Weight Assumption

The subgradient-push method

Every node i maintains auxiliary vector variables $\mathbf{x}_i(t), \mathbf{w}_i(t)$ in $\mathbb{R}^d$, as well as an auxiliary scalar variable $y_i(t)$, initialized as $y_i(0) = 1$ for all i. These quantities will be updated by the nodes according to the rules,

$$\begin{aligned}
\mathbf{w}_i(t+1) &= \sum_{j \in N_i^{\text{in}}(t)} \frac{\mathbf{x}_j(t)}{d_j(t)}, \\
y_i(t+1) &= \sum_{j \in N_i^{\text{in}}(t)} \frac{y_j(t)}{d_j(t)}, \\
\mathbf{z}_i(t+1) &= \frac{\mathbf{w}_i(t+1)}{y_i(t+1)}, \\
\mathbf{x}_i(t+1) &= \mathbf{w}_i(t+1) - \alpha(t+1)\mathbf{g}_i(t+1), \qquad (1)
\end{aligned}$$

where $\mathbf{g}_i(t+1)$ is a subgradient of the function f_i at $\mathbf{z}_i(t+1)$. The method is initiated with $\mathbf{w}_i(0) = \mathbf{z}_i(0) = \mathbf{1}$ and $y_i(0) = 1$ for all i. The stepsize $\alpha(t+1) > 0$ satisfies

the following decay conditions

$$\sum_{t=1}^{\infty} \alpha(t) = \infty, \qquad \sum_{t=1}^{\infty} \alpha^2(t) < \infty, \qquad \alpha(t) \leq \alpha(s) \text{ for all } t > s \geq 1. \tag{2}$$

We note that the above equations have simple broadcast-based implementation: each node i broadcasts the quantities $\mathbf{x}_i(t)/d_i(t), y_i(t)/d_i(t)$ to all of the nodes in its out-neighborhood, which simply sum all the messages they receive to obtain $\mathbf{w}_i(t+1)$ and $y_i(t+1)$. The update equations for $\mathbf{z}_i(t+1), \mathbf{x}_i(t+1)$ can then be executed without any further communications between nodes during step t.

We note that we make use here of the assumption that node i knows its out-degree $d_i(t)$.

Convergence

Our first theorem demonstrates the correctness of the subgradient-push method for an arbitrary stepsize $\alpha(t)$ satisfying Eq. (2).

Theorem 1 *Suppose that:*

(a) The graph sequence $\{G(t)\}$ is uniformly strongly connected with a self-loop at every node.

(b) Each function $f_i(\mathbf{z})$ is convex and the set $Z^* = \arg\min_{\mathbf{z}\in\mathbb{R}^d} \sum_{i=1}^m f_i(\mathbf{z})$ is nonempty.

(c) The subgradients of each $f_i(\mathbf{z})$ are uniformly bounded, i.e., there is $L_i < \infty$ such that

$$\|\mathbf{g}_i\|_2 \leq L_i \qquad \text{for all subgradients } \mathbf{g}_i \text{ of } f_i(\mathbf{z}) \text{ at all points } \mathbf{z} \in \mathbb{R}^d.$$

Then, the distributed subgradient-push method of Eq. (1) with the stepsize satisfying the conditions in Eq. (2) has the following property

$$\lim_{t\to\infty} \mathbf{z}_i(t) = \mathbf{z}^* \qquad \text{for all } i \text{ and for some } \mathbf{z}^* \in Z^*.$$

Convergence Rate

Our second theorem makes explicit the rate at which the objective function converges to its optimal value. As standard with subgradient methods, we will make two tweaks in order to get a convergence rate result:

(i) we take a stepsize which decays as $\alpha(t) = 1/\sqrt{t}$ (stepsizes which decay at faster rates usually produce inferior convergence rates),

(ii) each node i will maintain a convex combination of the values $\mathbf{z}_i(1), \mathbf{z}_i(2), \ldots$ for which the convergence rate will be obtained.

We then demonstrate that the subgradient-push converges at a rate of $O(\ln t/\sqrt{t})$. The result makes use of the matrix $A(t)$ that captures the weights used in the construction of $\mathbf{w}_i(t+1)$ and $y_i(t+1)$ in Eq. (1), which are defined by

$$A_{ij}(t) = \begin{cases} 1/d_j(t) & \text{whenever } j \in N_i^{\text{in}}(t), \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Convergence Rate

Theorem 2 *Suppose all the assumptions of Theorem 1 hold and, additionally,* $\alpha(t) = 1/\sqrt{t}$ *for* $t \geq 1$. *Moreover, suppose that every node* i *maintains the variable* $\widetilde{\mathbf{z}}_i(t) \in \mathbb{R}^d$ *initialized at time* $t = 1$ *to* $\widetilde{\mathbf{z}}_i(1) = \mathbf{z}_i(1)$ *and updated as*

$$\widetilde{\mathbf{z}}_i(t+1) = \frac{\alpha(t+1)\mathbf{z}_i(t+1) + S(t)\widetilde{\mathbf{z}}_i(t)}{S(t+1)},$$

where $S(t) = \sum_{s=0}^{t-1} \alpha(s+1)$. *Then, we have that for all* $t \geq 1$, $i = 1, \ldots, n$, *and any* $\mathbf{z}^* \in Z^*$,

$$\begin{aligned} F\left(\widetilde{\mathbf{z}}(t)\right) - F(\mathbf{z}^*) \;\leq\; & \frac{n}{2}\frac{\|\bar{\mathbf{x}}(0) - \mathbf{z}^*\|_1}{\sqrt{t}} + \frac{n}{2}\frac{\left(\sum_{i=1}^n L_i\right)^2}{4}\frac{(1+\ln t)}{\sqrt{t}} \\ & + \frac{16}{\delta(1-\lambda)}\left(\sum_{i=1}^n L_i\right)\frac{\sum_{j=1}^n \|\mathbf{x}_j(0)\|_1}{\sqrt{t}} + \frac{16}{\delta(1-\lambda)}\left(\sum_{i=1}^n L_i^2\right)\frac{(1+\ln t)}{\sqrt{t}} \end{aligned}$$

where

$$\bar{\mathbf{x}}(0) = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i(0),$$

and the scalars λ *and* δ *are functions of the graph sequence* $G(1), G(2), \ldots$, *which have the following properties:*

(a) For any B-connected graph sequence with a self-loop at every node,

$$\delta \geq \frac{1}{n^{nB}},$$
$$\lambda \leq \left(1 - \frac{1}{n^{nB}}\right)^{1/(nB)}.$$

(b) If each of the graphs $G(t)$ is regular then

$$\delta = 1$$
$$\lambda \leq \min\left\{\left(1 - \frac{1}{4n^3}\right)^{1/B}, \max_{t\geq 1}\sqrt{\sigma_2(A(t))}\right\}$$

where $A(t)$ is defined by Eq. (3) and $\sigma_2(A)$ is the second-largest singular value of a matrix A.

Several features of this theorem are expected: it is standard for a distributed subgradient method to converge at a rate of $O(\ln t/\sqrt{t})$ with the constant depending on the

S.S. Ram, A. Nedić, and V.V. Veeravalli, "Distributed Stochastic Subgradient Projection Algorithms for Convex Optimization," Journal of Optimization Theory and Applications,147 (3) 516–545, 2010

J.C. Duchi, A. Agarwal, and M.J. Wainwright, "Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling," IEEE Transactions on Automatic Control, 57(3) 592–606, 2012

subgradient-norm upper bounds L_i, as well as on the initial conditions $\mathbf{x}_i(0)$. Moreover, it is also standard for the rate to involve λ, which is a measure of the connectivity of the directed sequence $G(1), G(2), \ldots$; namely, the closeness of λ to 1 measures the speed at which a consensus process on the graph sequence $\{G(t)\}$ converges.

However, our bounds also include the parameter δ, which, as we will later see, is a measure of the imbalance of influences among the nodes. Time-varying directed regular networks are uniform in influence and will have $\delta = 1$, so that δ will disappear from the bounds entirely; however, networks which are, in a sense to be specified, non-uniform will suffer a corresponding blow-up in the convergence time of the subgradient-push algorithm.

The details are in:

AN and Alex Olshevsky, "Distributed optimization over time-varying directed graphs," http://arxiv.org/abs/1303.2289