# Distributed Optimization in Networked Systems

**Angelia Nedić**

*angelia@illinois.edu*

ISE Department and Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

Collaborators T. Başar, T-H. Chang, A.S. Morse, A. Olshevsky, A. Ozdaglar, P. Parrilo, A. Scaglione, U.V. Shanbhag, D. Stipanović, J. Tsitsiklis, V.V. Veeravali

Grad. students S.S. Ram, K. Srivastava, B. Touri, S. Lee, J. Koshal, R. Etesami

Postdoctorate researchers Ji Liu

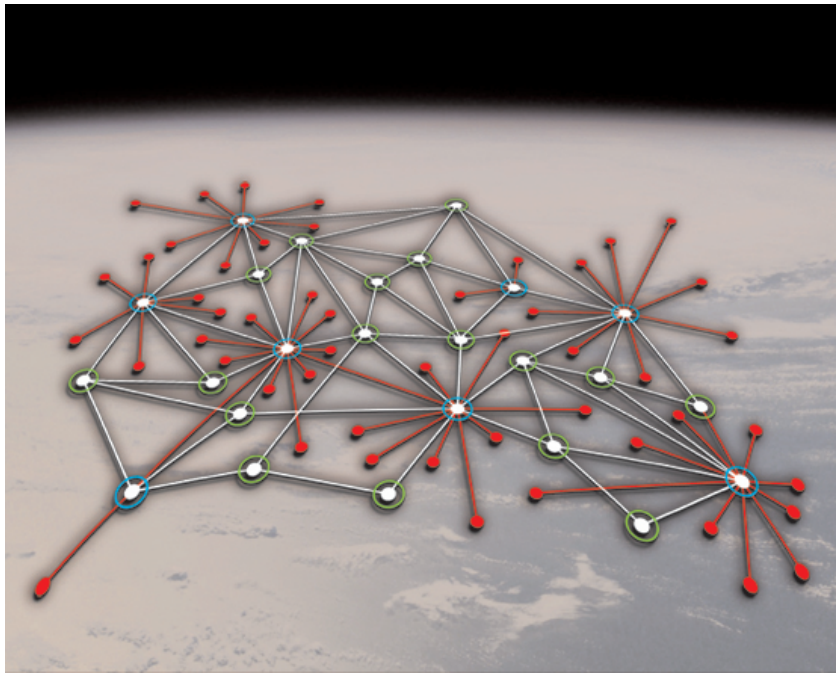






# Large Networked Systems

- 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

# New Applications - New Challenges

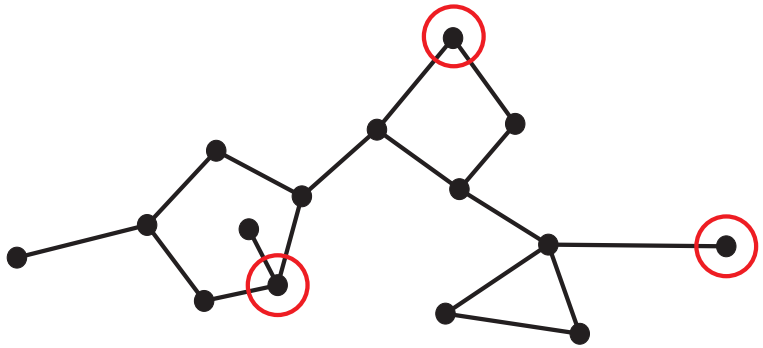
- **Lack of central "authority"**
  - The centralized network architecture is **not possible**
    - Size of the network / Proprietary issues
  - Sometimes the centralized architecture is **not desirable**
    - Security issues / Robustness to failures
- **Network dynamics**
  - Mobility of the network
    - The agent spatio-temporal dynamics
    - Network connectivity structure is varying in time
  - Time-varying network
    - The network itself is evolving in time
- The challenge is to control, coordinate, design protocols and analyze operations/performance over such networks

- **Goals:**

  Control-optimization algorithms deployed in such networks should be

  - Completely distributed relying on local information and observations
  - Robust against changes in the network topology
  - Easily implementable

# Example: Computing Aggregates in P2P Networks

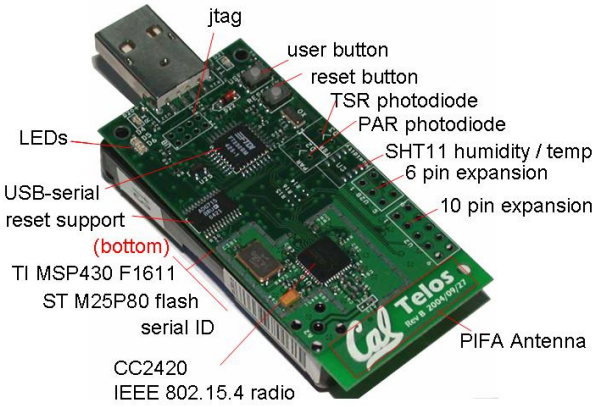


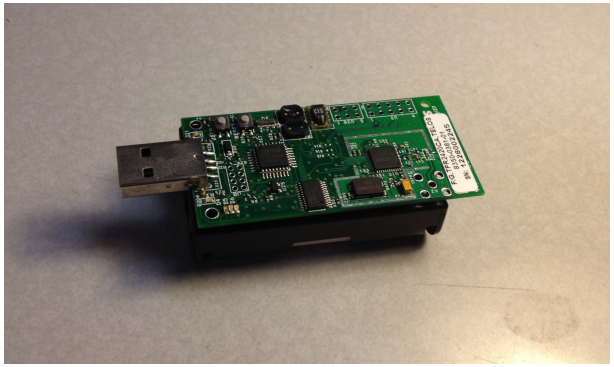
- Data network
  - Each node (location) $i$ has stored data/files with average size $\theta_i$
  - The value $\theta_i$ is known at that location only - no central access to all $\theta_i$, $i = 1, \ldots, m$
  - The nodes are connected over a static undirected network
- Distributedly compute the average size of the files stored?*
- Control/Game/Optimization Problem: **Agreement/Consensus Problem**

Optimization Formulation $\min_{x \in \mathbb{R}} \sum_{i=1}^{m} (x - \theta_i)^2$

*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, 2003.

# Sensor Networks

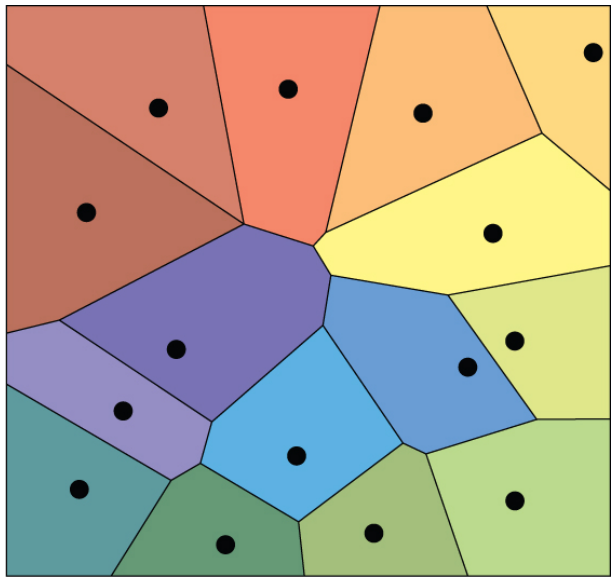


- A computing concept known as motes (smart dust, wireless sensing networks).
- They can be used in many different ways.
- For example, engineers may mix them into concrete to monitor the health of buildings and bridges (smart buildings), place them on power grids to monitor the power load (smart grids).
- It is a completely new paradigm for distributed sensing and it is opening up a fascinating new way to look at computers.

# Example in Sensor Networks: Determining Voronoi Cells

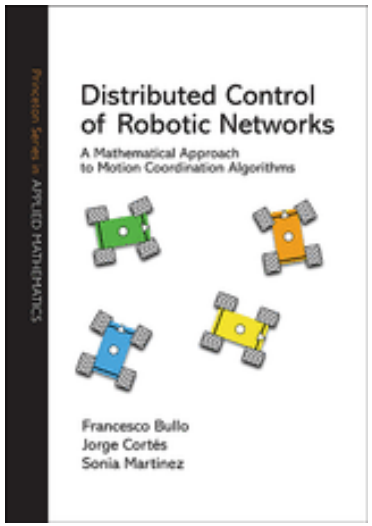
A Voronoi cell of a sensor in a network is the locus of points in a sensor field that are the closest to a given sensor among all other sensors (Bash and Desnoyers 07)

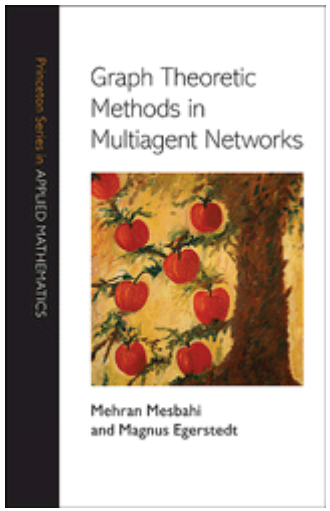


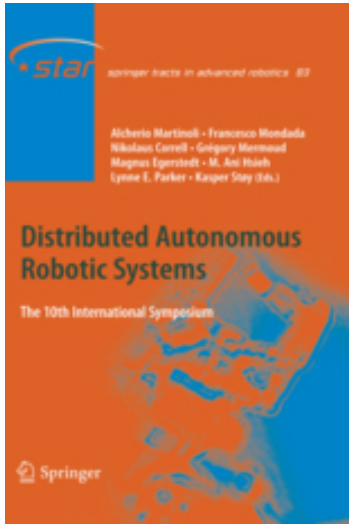
- A key building block supporting a number of applications in sensor networks (each sensor acts as a representative for the points in its cell)
  - Building a piece-wise approximation of the field
  - Multi-sensor target localization and tracking problems
  - Sensor data aggregation

# Many More Examples

- Bio-inspired systems
- Self-organized systems
- Social networks
- Opinion dynamics
- Averaging dynamics
- Multi-agent systems



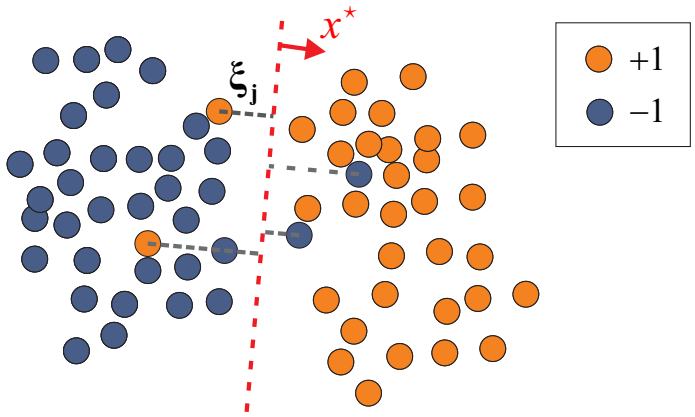



# Example: Support Vector Machine (SVM)

## Centralized Case

Given a data set $\{z_j, y_j\}_{j=1}^p$, where $z_j \in \mathbb{R}^d$ and $y_j \in \{+1, -1\}$



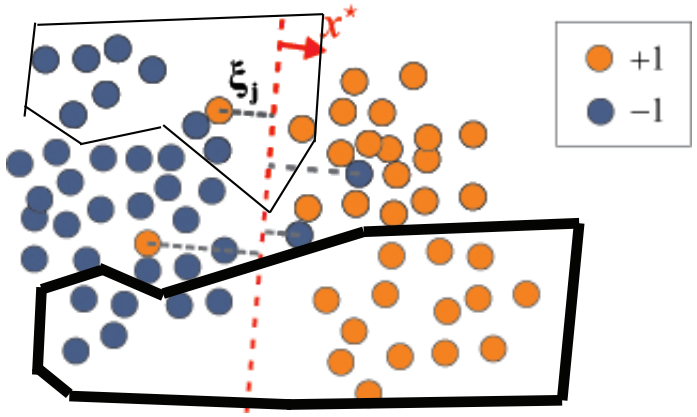

- Find a maximum margin separating hyperplane $x^\star$

  Centralized (not distributed) formulation

$$\min_{x \in \mathbb{R}^d, \xi \in \mathbb{R}^p} F(x) \triangleq \frac{\rho}{2}\|x\|^2 + \sum_{j=1}^{p} \max\{\xi_j, 1 - y_j\langle x, z_j\rangle\}$$

# Support Vector Machine (SVM) - Decentralized Case

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




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

$$\min_{x \in \mathbb{R}^d, \xi \in \mathbb{R}^p} \sum_{i=1}^{n} \left( \frac{\rho}{2n} \|x\|^2 + \sum_{j \in J_i} \max\{\xi_j, 1 - y_j \langle x, z_j \rangle\} \right)$$

$$\min_{x \in \mathbb{R}^d} F(x) = \sum_{i=1}^{n} f_i(x)$$

$$f_i(x) = \frac{\rho}{2n} \|x\|^2 + \sum_{j \in J_i} \min_{\xi_j \in \mathbb{R}} \max\{\xi_j, 1 - y_j \langle x, z_j \rangle\}$$

# Consensus Model

# Part I

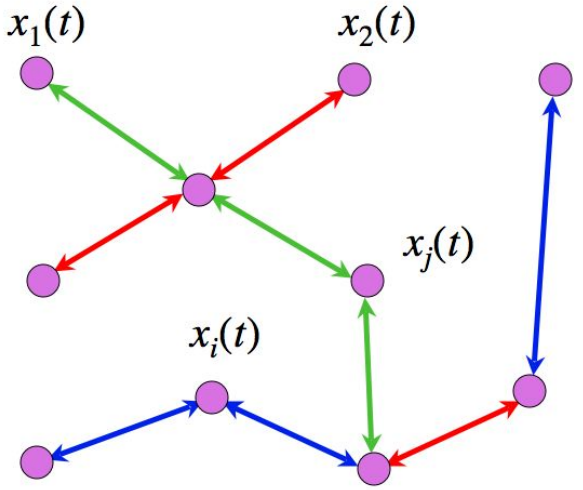
Network Diffusion Model/ Alignment Model

# 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.,

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




## Algorithm for Static Network Topology

each agent performs the following updates

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

where $N_i$ is the set of neighbors of agent $i$ in the network (including itself), and $a_{ij} > 0$ with $\sum_{j\in N_i} a_{ij} = 1$.

# Dynamic Network Topology

Each agent dynamic is given by

$$x_i(k+1) = \sum_{j \in \mathcal{N}_i(k)} a_{ij}(k) x_j(k)$$

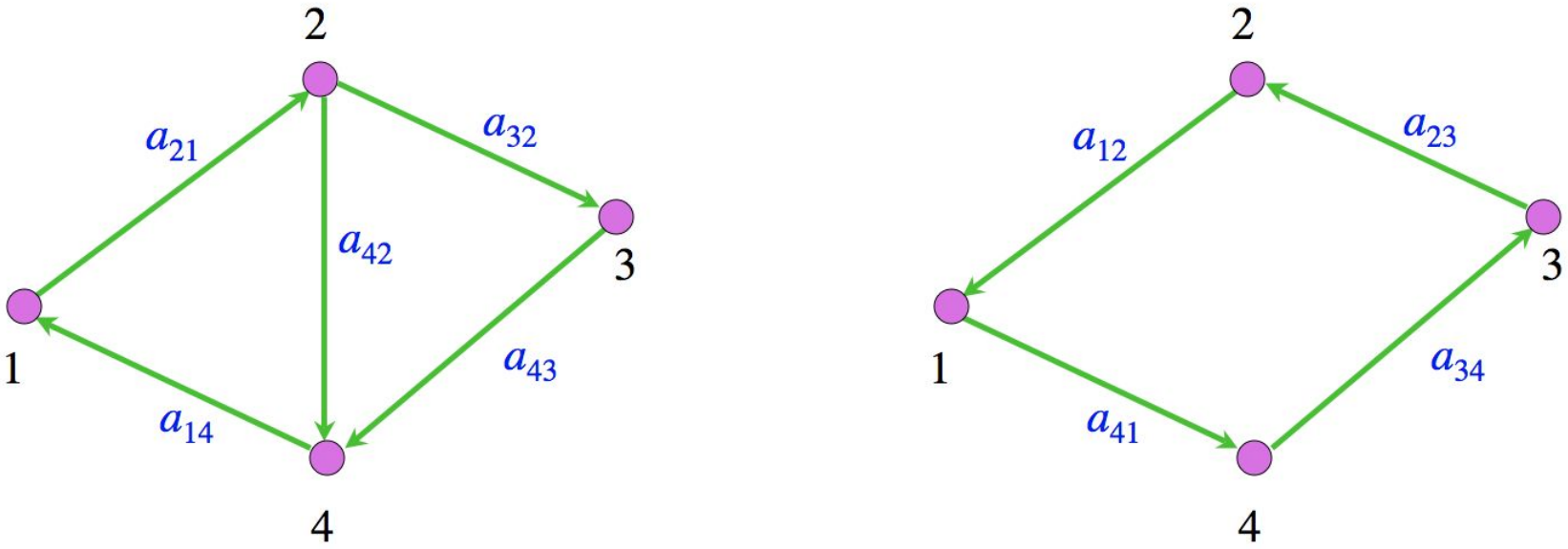
where $N_i(k)$ is the set of neighbors of agent $i$ (including itself) and $a_{ij}(k)$ are the weights that agent $i$ assigns to its neighbors at time $k$.

- The time is slotted $t_0, t_1, \ldots$; time $k$ corresponds to slot $k$
- The set $N_i(k)$ of neighbors is changing with time
- The weights $a_{ij}(k)$ are changing with time
- **The weights are nonnegative and sum to 1**

$$a_{ij}(k) > 0, \ j \in N_i(k) \quad \text{and} \quad \sum_{j \in N_i(k)} a_{ij}(k) = 1 \qquad \text{for all } i \text{ and } k$$

# Connectivity Graphs

The connectivity graph $(V, \mathcal{E}_k)$ is the directed graph with node set $V = \{1, \ldots, m\}$ and the edge set $\mathcal{E}_k = \{(i, j) \mid i \text{ receives information from } j \text{ at time } k\}$.




Examples of the connectivity graphs at times 1 and 2, resp. from the left to the right

$$\mathcal{E}_1 = \{(1,4),(2,1),(3,2),(4,2),(4,3)\}$$
$$\mathcal{E}_2 = \{(1,2),(2,3),(3,4),(4,1)\}$$

# Weight Matrices

Introduce the weight matrix $A(k)$ which is compliant with the connectivity graph $(V, \mathcal{E}_k)$ enlarged with the self-loops:

$$a_{ij}(k) = \begin{cases} a_{ij}(k) > 0 & \text{if either } (i,j) \in \mathcal{E}_k \text{ or } j = i \\ 0 & \text{otherwise} \end{cases}$$

**Assumption 1:** For each $k$,

- **The graph $(V, \mathcal{E}_k)$ is strongly connected** (there is a directed path from each node to every other node in the graph).

- **The matrix $A(k)$ is row-stochastic** (it has nonnegative entries that sum to 1 in each row).

- **The positive entries of $A(k)$ are uniformly bounded away from zero**: for a scalar $\eta > 0$ and for all $i, j, k$

$$\text{if } a_{ij}(k) > 0 \qquad \text{then} \qquad a_{ij}(k) \geq \eta.$$

# Basic Result

**Proposition**[†] Under Assumption 1, the agent values converge to a consensus with a geometric rate. In particular,

$$\lim_{k\to\infty} x_i(k) = \alpha \qquad \text{for all } i,$$

where $\alpha$ is some convex combination of the initial values $x_1(0), \ldots, x_m(0)$; i.e., $\alpha = \sum_{j=1}^m \pi_j x_j(0)$ with $\pi_j > 0$ for all $j$, and $\sum_{j=1}^m \pi_j = 1$.
Furthermore

$$\max_i x_i(k) - \min_j x_j(k) \leq \left(\max_i x_i(0) - \min_j x_j(0)\right) \beta^{\frac{k}{m-1}} \qquad \text{for all } k,$$

where $\beta = 1 - m\eta^{m-1}$.

**The convergence rate is geometric**

[†] Tsitsiklis 1984, AN and Ozdaglar 2010.

# Proof Outline

- Write the system compactly as

$$x(k+1) = A(k)x(k)$$

and consider the function

$$V(k) = \max_i x_i(k) - \min_j x_j(k)$$

as a "measure" of progress toward the consensus (it can be viewed as "the size of disagreement" among the agents at time $k$).

- Under the stochasticity of the weights $A(k)$, the function $V(k)$ is nonincreasing in time.

- Further, consider the behavior of the system over a window of time

$$x(k+1) = \Phi(k,s)x(s), \qquad \Phi(k,s) = A(k)A(k-1)\cdots A(s+1)A(s).$$

- Under the connectivity and weight conditions of Assumption 1, the matrices $\Phi((k+1)B-1, kB)$ are *primitive*[‡] for some positive integer $B$ that depends on $m$

[‡] $A$ is primitive when $a_{ij} > 0$ for all entries

- This implies a strict "significant" decrease from $kB$ to $(k+1)B$ for the function $V(k) = \max_i x_i(k) - \min_j x_j(k)$:
  the difference $V(kB) - V((k+1)B)$ can be bounded by a constant term independent of $k$, but dependent on $B$.

- This, and the nonincreasing property of $V(k)$ yield the convergence and the convergence rate estimate.

# Average Consensus

We want agents to agree on the average of their initial values, i.e., we want

$$\lim_{k\to\infty} x_i(k) = \frac{1}{m}\sum_{j=1}^{m} x_j(0) \qquad \text{for all } i.$$

This will happen when the weight matrices $A(k)$ are doubly stochastic each (each row and each column sum is equal to 1).

**Proposition 3** [Nedić, Olshevsky, Ozdaglar, Tsitsiklis 09] Under Assumption 1 and the doubly stochasticity of the weights, the agent values converge to the average consensus with a geometric rate. Specifically,

$$\lim_{k\to\infty} x_i(k) = \frac{1}{m}\sum_{j=1}^{m} x_j(0) \qquad \text{for all } i,$$

$$\left| x_i(k) - \frac{1}{m}\sum_{j=1}^{m} x_j(0) \right| \leq c\left(1 - \frac{\eta}{4m^2}\right)^k \qquad \text{for all } i \text{ and } k,$$

where the constant $c > 0$ depends on $\eta$ and $m$.

The importance of the preceding result is in a *polynomial dependence* of the bound on the size $m$ of the network (as opposed to *exponential* which is known for the general consensus problem).

**Proof Outline**

- Write the system compactly as: $x(k+1) = A(k)x(k)$ and consider the function

$$V(k) = \sum_{i=1}^{m} \left( x_i(k) - \frac{1}{m} \sum_{j=1}^{m} x_j(k) \right)^2$$

  as a "measure" of progress toward the consensus (empirical variance among the agent values at time $k$).
- Under the doubly stochasticity of $A(k)$, the average does not change, $\sum_{j=1}^{m} x_j(k) = \sum_{j=1}^{m} x_j(0)$, and function $V(k)$ is nonincreasing in time.
- Under Assumption 1, using certain cuts in the graphs, the difference $V(k) - V(k+1)$ can be bounded by a constant term.
- This yields both the convergence and the convergence rate estimate.

# Further Results

- The consensus and the average consensus results hold when the connectivity assumption at each time instant $k$ is replaced by a **connectivity over a bounded time interval** Nedić, Olshevsky, Ozdaglar and Tsitsiklis 09 (NOOT09)
- Consensus in the presence of **delays** (Nedić and Ozdaglar 08, Bliman, Nedić and Ozdaglar)
- **Quantization** effects (Kashyap, Srikant and Başar 06, Carli *et al.* 07, Kar and Moura 07, NOOT09)
- Consensus over **random networks** (Tahbaz-Salehi and Jadbabaie 08, Hatano and Mesbahi 05, Touri 11)
- Consensus over a network with **noisy links** (Kar and Moura 07, Touri and Nedić 09)
- Consensus with various aspects: A.S. Morse, J. Lin, B.D.O. Anderson, M. Cao, B. Touri, W. Rei, D. Shah, A. Scaglione, M. Rabbat, M. Johansson, K. Johansson, J. Lorenz, J. Cortes, L. Pavel, K. Kvaternik, M. Hong, B. Gharesifard, A. Dominguez-Garcia, C. Hadjicostis, S. Sundaram, N. Vaidya, . . .

# Consensus as Optimization Problem: Static Network

- Given a connected graph and a stochastic weight matrix $A$ (positive diagonal)

- Consensus problem: determining an $x$ such that

$$Ax = x$$

- This is a **Fixed-Point Problem**: under certain "contractive" properties of $A$, fixed-point iterations

$$x_{k+1} = Ax_k$$

converge to a fixed point.

- **Merit-Function Approach**:

$$\min_{x \in \mathbb{R}^m} \|Ax - x\|^2$$

- Consensus algorithms can be viewed are distributed algorithms for minimization of some "merit function" associated with the underlying fixed-point problem

# Consensus as Optimization Problem: Dynamic Network

- Given a sequence $\{G_t\}$ of connected graphs and a sequence of stochastic weight matrices $\{A(t)\}$
- Consensus problem: determining an $x$ such that

$$A(t)x = x \qquad \text{for all } t$$

- This is a **Fixed-Point Problem**: under certain "contractive" properties of $A(t)$, fixed-point iterations

$$x_{t+1} = A(t)x_t$$

converge to a fixed point.
**NOTE:** all matrices $A(t)$ have a set of common fixed points!

- **Merit-Function Approach**:

$$\min_{x\in\mathbb{R}^m} \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \|A(t)x - x\|^2$$

- Special property in consensus problems:

**The functions in the sum have a common set of global minima**

# Consensus as Optimization Problem: Laplacians

- Given a graph and a weight matrix $A$

- Letting $L_A$ be the Laplacian associated with $(G, A)$ consensus problem: determining an $x$ such that

$$L_A x = 0$$

- This is a **Zero-Point Problem** that is moved to a fixed point problem of the form

$$(I - \alpha L_A)\, x = x \qquad \text{for any } \alpha.$$

where $I$ is the identity.

- Similarly for time-varying graphs

## Computational Model

# Part II

## Distributed Optimization in Network

- Optimization problem - classic

- Problem data distributed - new

# 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 the whole system information**

- MAIN IDEA: use local information exchange to spread the information through the entire network

  **use network to diffuse information - virtual global coordinator**

- 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:

    **synchonous communications**

    **asynchronous: gossip or broadcast communications**

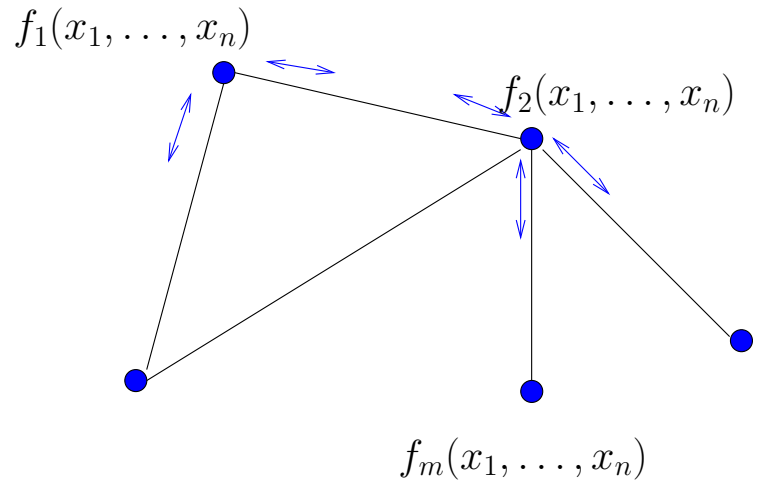
# General Multi-Agent Model

- 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

The problem can be formalized:

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

**Distributed Self-organized Agent System**



# How Agents Manage to Optimize Global Network Problem?

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

- 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.

# Distributed Optimization Algorithm

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

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

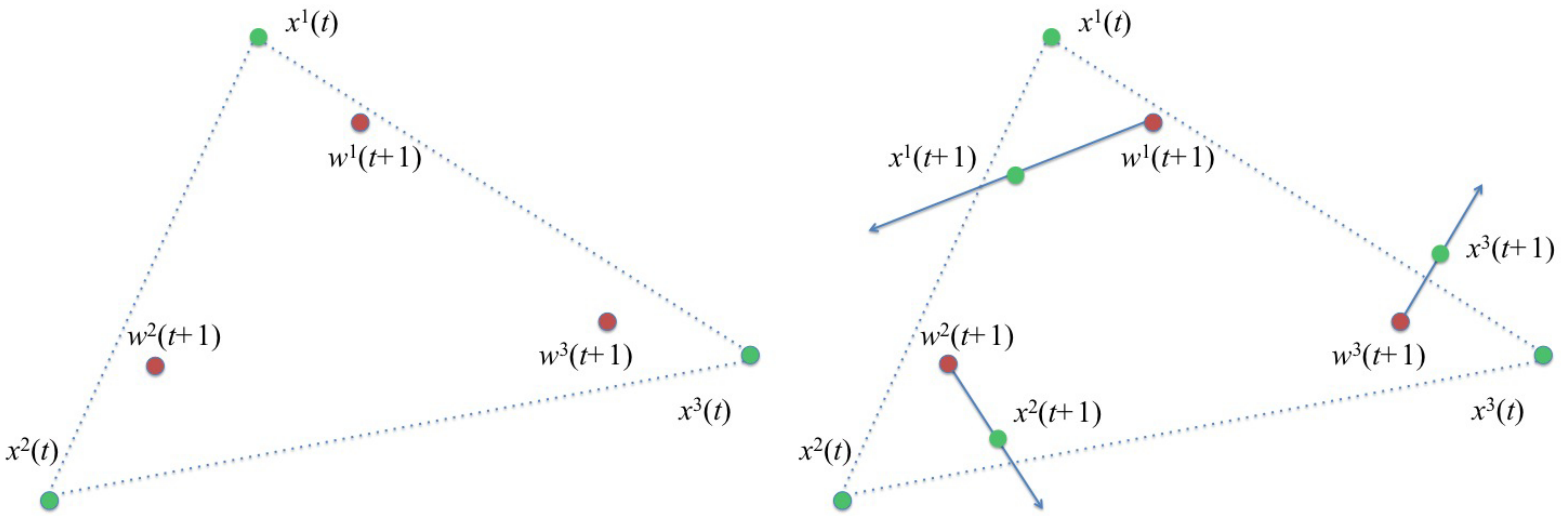
- At time $t+1$, agents communicate their estimates to their neighbors and update by performing two steps:
  - **Consensus-like step** to mix their own estimate with those received from neighbors

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

  - 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

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

For convergence it suffices two relations to hold:

- $\sum_{t=0}^\infty \alpha(t)C \sum_{i=1}^m \|s(t+1) - w_i(t+1)\| < \infty$, which allows us to conclude
$$F(s(t+1)) - F(x^*) \to 0 \qquad \text{along a subsequence}$$
$F(x) = \sum_{i=1}^m f_i(x)$

- But this is not enough, we also want to show that $x_i(t)$ converge to the same optimal solution for all agents $i$. This will hold if we can show
$$\|s(t+1) - x_i(t+1)\| \to 0 \text{ as } t \to \infty \text{ for all } i$$

The trouble is in showing $\|s(t+1) - x_i(t+1)\| \to 0$ 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,

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

By choosing $\tau$ 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

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

# 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.N, 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

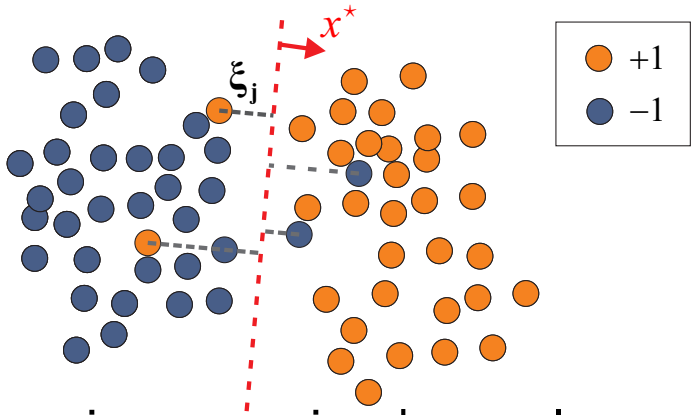
K. Srivastava and A. Nedić "Distributed Asynchronous Constrained Stochastic Optimization," *IEEE Journal of Selected Topics in Signal Processing* 5 (4) 772-790, 2011.

¶Uses different weights

# Revisited Example: Support Vector Machine (SVM)

## Centralized Case

Given a data set $\{(z_j, y_j),\ j = 1, \ldots, p\}$, where $z_j \in \mathbb{R}^d$ and $y_j \in \{+1, -1\}$




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

  Centralized (not distributed) formulation

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

$$\text{s.t. } (x, \xi) \in X \triangleq \{(x, \xi) \mid y_j \langle x, z_j \rangle \geq 1 - \xi_j, \xi_j \geq 0,\ j = 1, \ldots, p\}$$

# Often Reformulated as: Data Classification

Given a set of data points $\{(z_j, y_j), j = 1, \ldots, p\}$, find a vector $(x, u)$ that

$$\text{minimizes} \quad \frac{\lambda}{2}\|x\|^2 + \sum_{j=1}^{p} \max\{0, 1 - y_j(\langle x, z_j\rangle + u)\}$$

Suppose that the data is distributed at $m$ locations, with each location having data points $\{(z_\ell, y_\ell), \ell \in S_i\}$, with $S_i$ being the index set

The problem can be written as:

$$\text{minimize} \sum_{i=1}^{m} \underbrace{\left( \frac{\lambda}{2m}\|x\|^2 + \sum_{\ell \in J_i} \max\{0, 1 - y_\ell(\langle x, z_\ell\rangle + u)\} \right)}_{f_i(\mathbf{x})} \quad \text{over } \mathbf{x} = (x, u) \in \mathbb{R}^n \times \mathbb{R}$$

Distributed algorithm has the form:

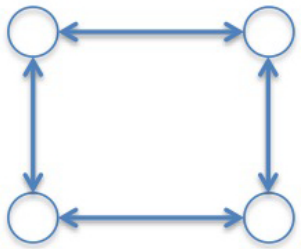
$$w_i(t+1) = \mathbf{x}_i(t) - \eta(t) \sum_{j=1}^{m} r_{ij} \mathbf{x}_j(t) \qquad (r_{ij} = 0 \text{ when } j \notin N_i)$$

$$\mathbf{x}_i(t+1) = w_i(t+1) - \alpha(t) \underbrace{g_i(w_i(t+1))}_{\text{subgradient of } f_i}$$

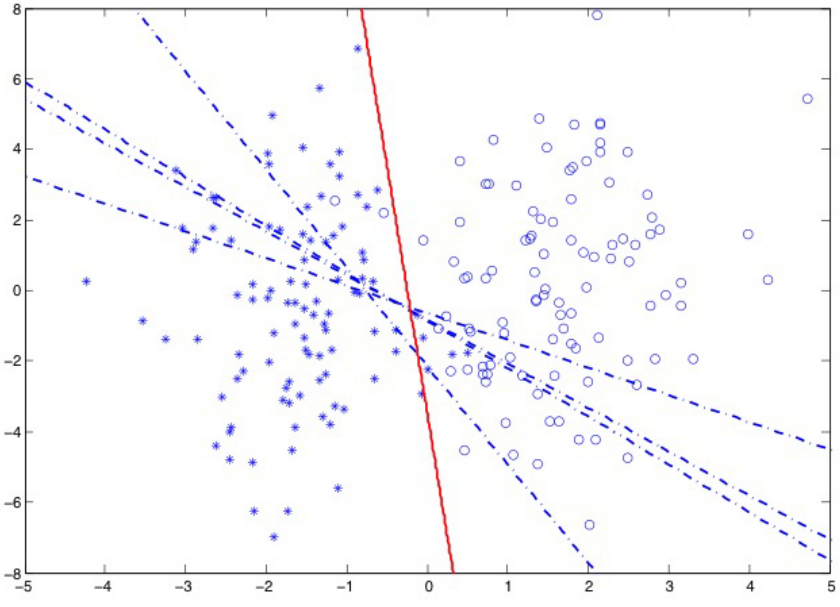
Algorithm is discussed in K. Srivastava and AN "Distributed Asynchronous Constrained Stochastic Optimization" *IEEE Journal of Selected Topics in Signal Processing* 5 (4) 772-790, 2011.

# Case with perfect communications

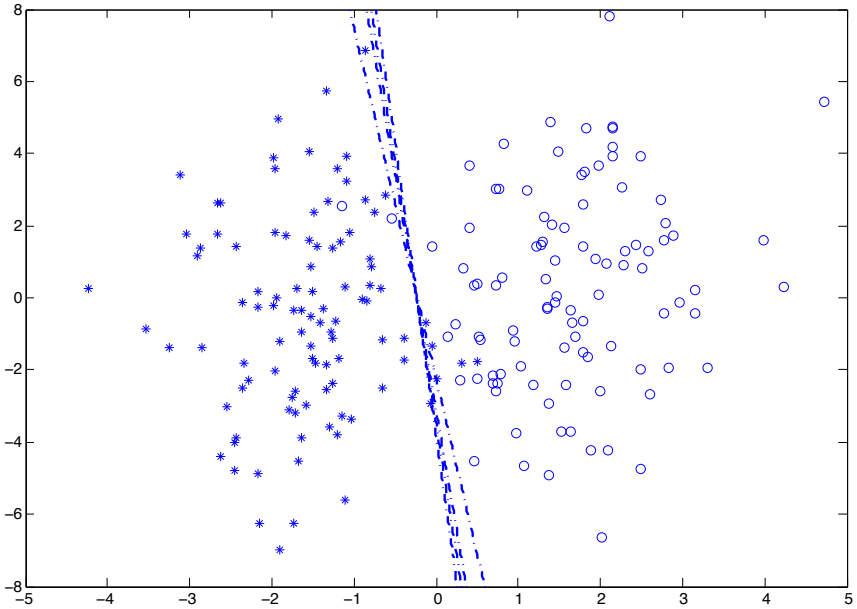
Illustration uses a simple graph of 4 nodes organized in a ring-network



$$\lambda = 6$$
$$\alpha(t) = \frac{1}{t}$$
$$\eta(t) = 0.8$$



After 20 iterations



After 500 iterations

# Case with imperfect communications

$$\text{minimize} \sum_{i=1}^{m} \underbrace{\left( \frac{\lambda}{2m}\|x\|^2 + \sum_{\ell \in J_i} \max\{0, 1 - y_\ell(\langle x, z_\ell \rangle + u)\} \right)}_{f_i(\mathbf{x})} \quad \text{over } \mathbf{x} = (x, u) \in \mathbb{R}^n \times \mathbb{R}$$

$$w_i(t+1) = \mathbf{x}_i(t) - \eta(t) \sum_{j=1}^{m} r_{ij}(\mathbf{x}_j(t) + \underbrace{\xi_{ij}(t)}_{\textbf{noise}})$$

with $r_{ij} = 0$ when $j \notin N_i$, $\eta(t) > 0$ is a noise-damping stepsize

$$\mathbf{x}_i(t+1) = w_i(t+1) - \alpha(t) g_i(w_i(t+1))$$

Noise-damping stepsize $\eta(t)$ has to be coordinated with sub-gradient related stepsize $\alpha(t)$

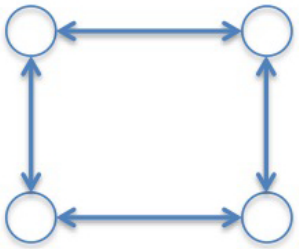
$$\sum_t \alpha(t) = \infty, \qquad \sum_t \alpha^2(t) < \infty$$

$$\sum_t \eta(t) = \infty, \qquad \sum_t \eta^2(t) < \infty$$

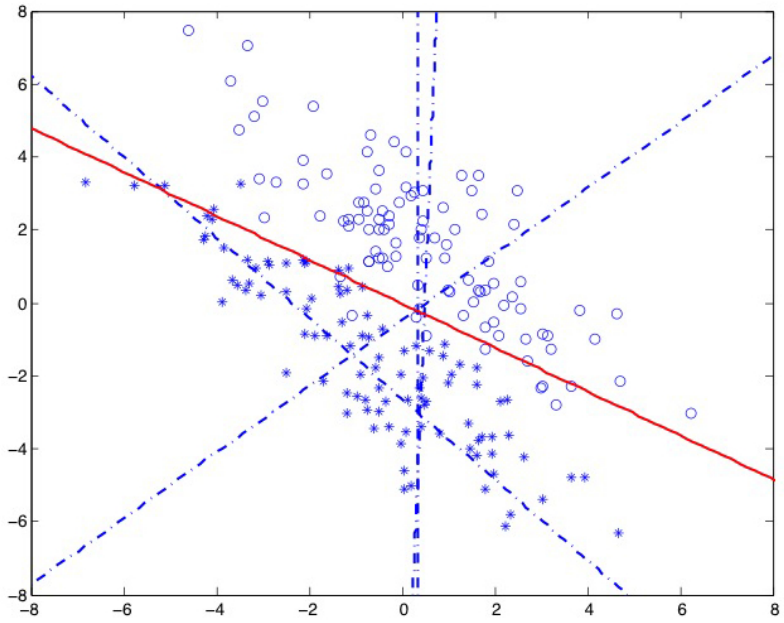
$$\sum_t \alpha(t)\eta(t) < \infty, \qquad \sum_t \frac{\alpha^2(t)}{\eta(t)} < \infty$$

# Case with imperfect communications

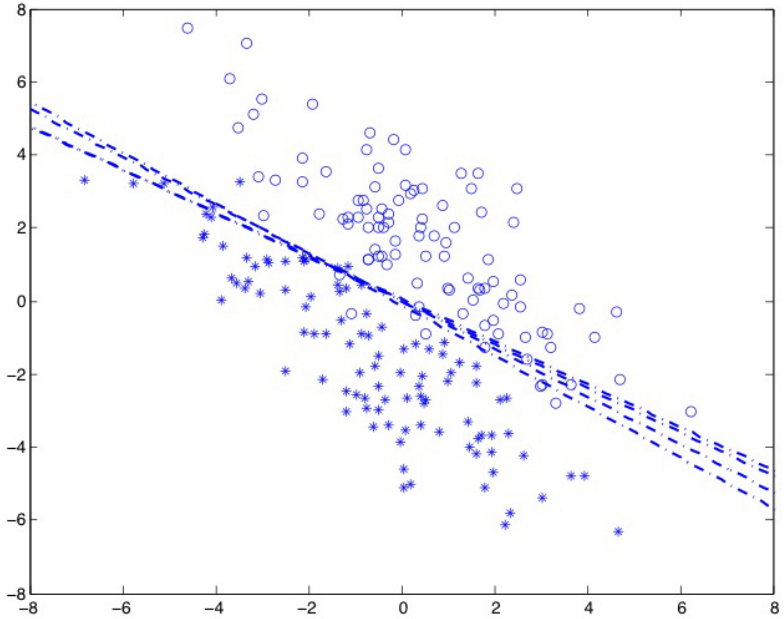
Illustration uses a simple graph of 4 nodes organized in a ring-network



$$\lambda = 6$$
$$\alpha(t) = \frac{1}{t}$$
$$\eta(t) = \frac{1}{t^{0.55}}$$



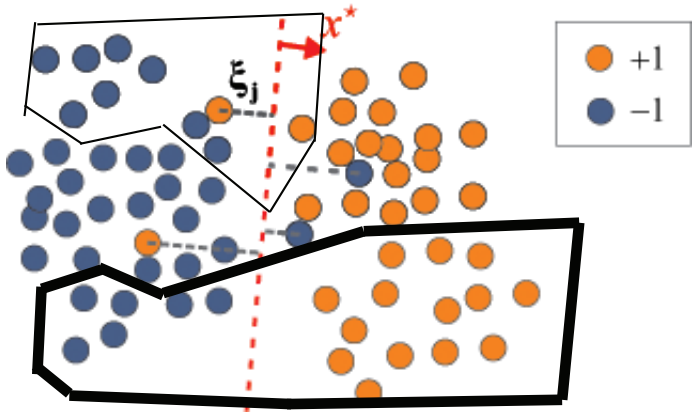
After 1 iteration



After 500 iterations

# RETURN TO ORIGINAL FORMULATION: Support Vector Machine (SVM) - Decentralized Case

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




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

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

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

$$X_i \triangleq \{(x,\xi) \mid y_j\langle x, z_j\rangle \geq 1-\xi_j, \xi_j \geq 0,\ j \in J_i\} \qquad i = 1,\ldots,m$$

# Challenge with Support Vector Machine (SVM)

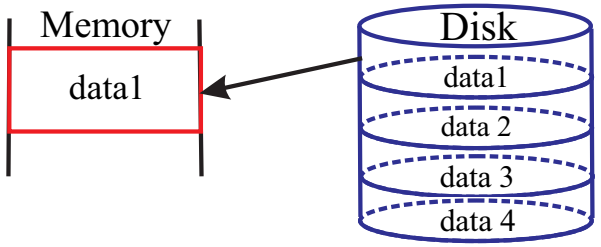
- **Challenge 1**: Online Learning (the constraint set $X$ not known in advance)
  - Standard gradient projection cannot be used

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

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



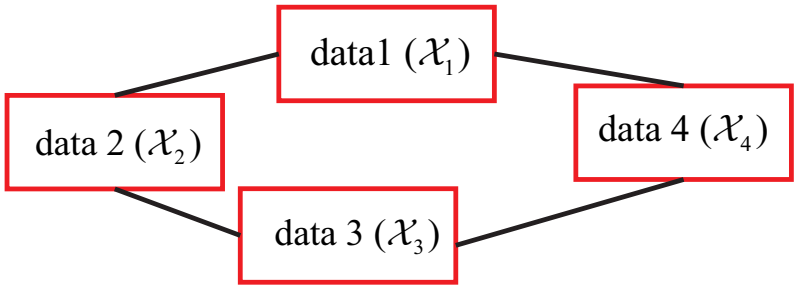
- Distributed formulation

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

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

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

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

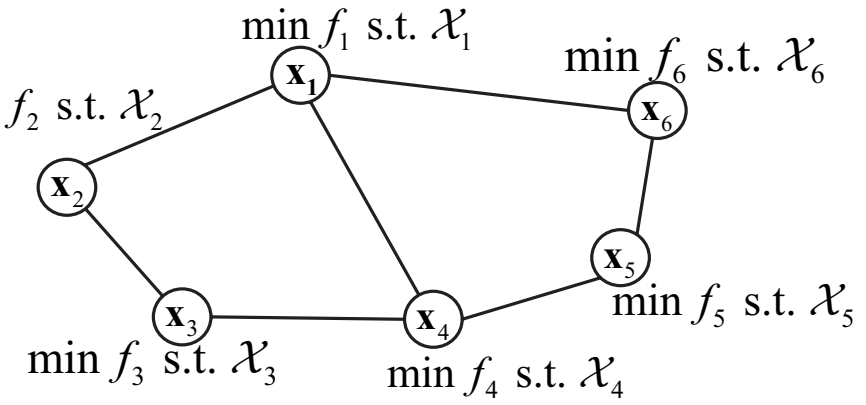



- **More efficient** if communication cost is low

# 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
  - $\min F(x) = \sum_{i=1}^{m} f_i(x) \qquad \text{s.t. } x \in X \triangleq \bigcap_{i=1}^{m} X_i, \; X_i = \cap_{j \in I_i} X_i^j$

# Existing Distributed Optimization Models

- Jie Lu and Choon Yik Tang
- **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

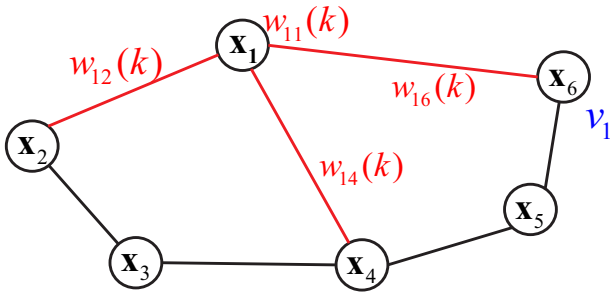
[‡‡] K. Srivastava and A. Nedić, "Distributed asynchronous constrained stochastic optimization," IEEE J. Sel. Topics. Signal Process., vol. 5, no. 4, pp. 772–790, 2011.

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 $i$ performs the following steps

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

  **2.** **Gradient update** $\quad \tilde{v}_i(k) = v_i(k) - \alpha_k \nabla f_i(v_i(k))$

  **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.

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

---

S. Lee and AN "Distributed Random Projection Algorithm for Convex Optimization" JSTSP 2013

S. Lee "Asynchronous Gossip-Based Random Projection Algorithms Over Networks," submitted 2013

# 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

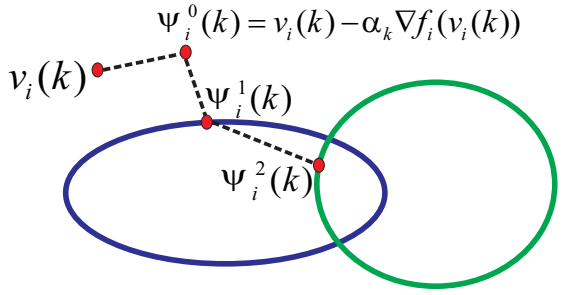
**1. Mixing** $v_i(k) = \sum_{j=1}^m a_{ij}(k) x_j(k)$

**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 $\Omega_i^r(k) \in J_i$, $r = 1, \ldots, b$ is drawn. The components $X_i^{\Omega_i^1(k)}, \ldots, X_i^{\Omega_i^b(k)}$ of $X_i = \bigcap_{j \in J_i} X_i^j$ 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 $X_i^j$, $j \in J_i$ 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.

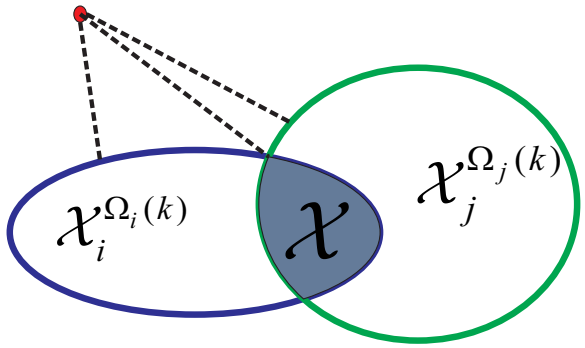
# Assumption on the Set Process $\{\Omega_i(k)\}$

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

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

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

(d) $\sum_{i=1}^{m} [A(k)]_{ij} = 1$ 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

| 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: $a_{ij}(k) = \frac{1}{|N_i(k)|}$

# 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 $C11$

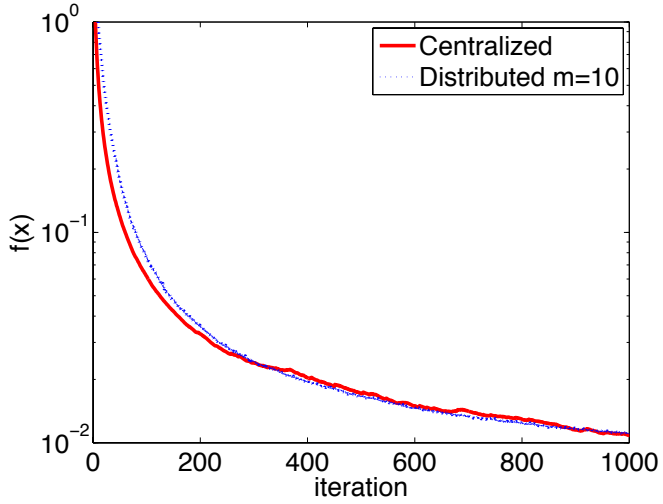
# Simulation Results

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

| Data set | $b$ | **Clique, $m = 10$** | $\bar{\mu}$ | $\bar{\sigma}$ | 95% confidence |
|---|---|---|---|---|---|
| `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] |
| `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] |
| `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 $F(x)$ (centralized) and case when $m = 10$ with $b = 1$

# Proof Sketch: Projection Error

Lemma 1: Projection Error

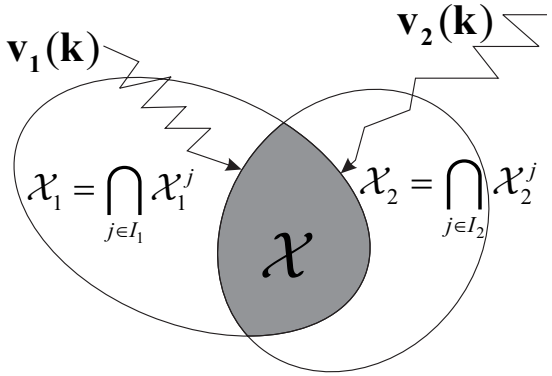
Let Assumption 1-3 hold. Let $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$.

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: Disagreement Estimate

Lemma 2: Disagreement Estimate

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

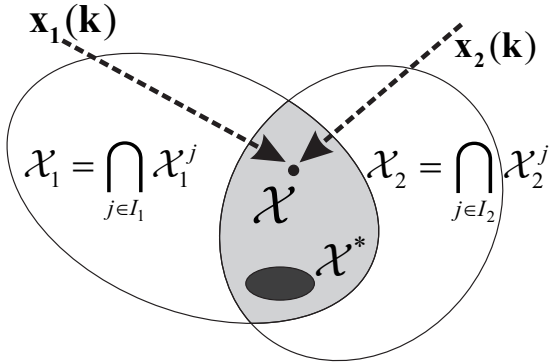
$$x_i(k+1) = \sum_{j=1}^{m} [W(k)]_{ij} x_j(k) + e_i(k) \quad \text{for all } i.$$

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

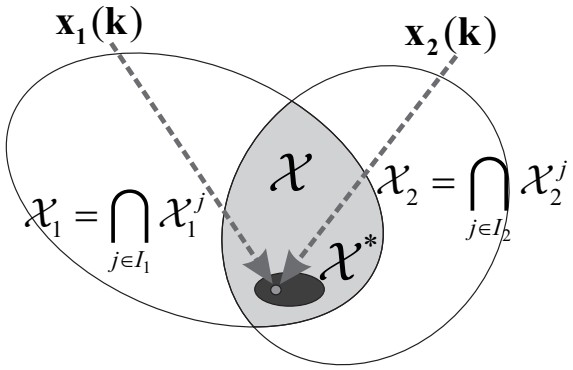
Then, for all $i, j$,

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

- 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: Almost-Supermartingale Relation




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: Combine All

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

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

# Advantages/Disadvatages

- Network can be used to diffuse information to all the nodes in that is not "globally available"
- The speed of the information spread depends on networks connectivity as well as communication protocols that are employed
- Mixing can be slow but it is stable
- Error/rate estimates are available and scale as $m^{3/2}$ at best in the size $m$ of the network
- Problems with special structure - may have better rates - Jakovetić, Xavier, Moura†
- Drawback: Doubly stochastic weights are required:
  - Can be accomplished with some additional "weights" exchange in bi-directional graphs
  - Difficult to ensure in directed graphs¶

† D. Jakovetić, J. Xavier, J. Moura "Distributed Gradient Methods" arxiv 2011

¶B. Gharesifard and J. Cortes, "Distributed strategies for generating weight-balanced and doubly stochastic digraphs," European Journal of Control, 18 (6), 539-557, 2012

## Push-Sum Based Computational Model

# Part III

Distributed Optimization in Directed Networks

# Model without Doubly Stochastic Weights

## Joint recent work with A. Olshevsky

### Push-Sum Model for Consensus for Time-Varying Directed Graphs

Every node $i$ maintains scalar variable $\mathbf{x}_i(t)$ and $y_i(t)$

These quantities will be updated by the nodes according to the rules,

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

- Each node $i$ "knows" its out degree $d_i(t)$ (includes itself) at every time $t$
- $N_i^{\text{in}}(t)$ is the "in"-degree of node $i$ at time $t$
- The method[†] is initiated with $\mathbf{w}_i(0) = \mathbf{z}_i(0) = \mathbf{1}$ and $y_i(0) = 1$ for all $i$.

D. Kempe, A. Dobra, and J. Gehrke "Gossip-based computation of aggregate information" In Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, pages 482491, Oct. 2003

F. Benezit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli "Weighted gossip: distributed averaging using non-doubly stochastic matrices" In Proceedings of the 2010 IEEE International Symposium on Information Theory, Jun. 2010.

# Convergence Result

Consider the sequences $\{z_i(t)\}$, $i = 1, \ldots, m$, generated by the push-sum method. Assuming that the graph sequence $\{G(t)\}$ is $B$-uniformly strongly connected, the following statements hold: For all $t \geq 1$ we have

$$\left| z_i(t+1) - \frac{\mathbf{1}'x(t)}{n} \right| \leq \frac{8}{\delta} \left( \lambda^t \|x(0)\|_1 + \sum_{s=1}^{t} \lambda^{t-s} \|\epsilon(s)\|_1 \right),$$

where $\delta > 0$ and $\lambda \in (0, 1)$ satisfy

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

Define matrices $A(t)$ by $A_{ij}(t) = 1/d_j(t)$ for $j \in N_i^{\text{in}}(t)$ and 0 otherwise

If each of the matrices $A(t)$ are doubly stochastic, then

$$\delta = 1, \ \lambda \leq \left\{ \left(1 - \frac{1}{4n^3}\right)^{1/B}, \max_{t \geq 0} \sqrt{\sigma_2(A(t))} \right\}.$$

# Optimization

**The subgradient-push method can be used for minimizing $F(z) = \sum_{i=1}^{m} f_i(z)$ over $z \in \mathbb{R}^d$**

Every node $i$ maintains scalar variables $\mathbf{x}_i(t), \mathbf{w}_i(t)$ in $\mathbb{R}$, 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,

$$\begin{aligned}
\mathbf{w}_i(t+1) &= \sum_{j \in N_i^{\text{in}}(t)} \frac{\mathbf{x}_j(t)}{d_j(t)}, \\
y_i(t+1) &= \sum_{j \in N_i^{\text{in}}(t)} \frac{y_j(t)}{d_j(t)}, \\
\mathbf{z}_i(t+1) &= \frac{\mathbf{w}_i(t+1)}{y_i(t+1)}, \\
\mathbf{x}_i(t+1) &= \mathbf{w}_i(t+1) - \alpha(t+1)\mathbf{g}_i(t+1), \qquad (2)
\end{aligned}$$

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

$$\sum_{t=1}^{\infty} \alpha(t) = \infty, \qquad \sum_{t=1}^{\infty} \alpha^2(t) < \infty, \qquad \alpha(t) \leq \alpha(s) \text{ for all } t > s \geq 1. \tag{3}$$

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 $d_i(t)$.

# Related Work: Static Network


- A.D. Dominguez-Garcia and C. Hadjicostis. Distributed strategies for average consensus in directed graphs. In Proceedings of the IEEE Conference on Decision and Control, Dec 2011.
- C. N. Hadjicostis, A.D. Dominguez-Garcia, and N.H. Vaidya, "Resilient Average Consensus in the Presence of Heterogeneous Packet Dropping Links" CDC, 2012
- K.I. Tsianos. The role of the Network in Distributed Optimization Algorithms: Convergence Rates, Scalability, Communication / Computation Tradeoffs and Communication Delays. PhD thesis, McGill University, Dept. of Electrical and Computer Engineering, 2013.
- K.I. Tsianos, S. Lawlor, and M.G. Rabbat. Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning. In Proceedings of the 50th Allerton Conference on Communication, Control, and Computing, 2012.
- K.I. Tsianos, S. Lawlor, and M.G. Rabbat. Push-sum distributed dual averaging for convex optimization. In Proceedings of the IEEE Conference on Decision and Control, 2012.
- K.I. Tsianos and M.G. Rabbat. Distributed consensus and optimization under communication delays. In Proc. of Allerton Conference on Communication, Control, and Computing, pages 974982, 2011.

# Convergence

Our first theorem demonstrates the correctness of the subgradient-push method for an arbitrary stepsize $\alpha(t)$ satisfying Eq. (3).

**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. (2) with the stepsize satisfying the conditions in Eq. (3) 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. (2), which are defined by

$$A_{ij}(t) = \begin{cases} 1/d_j(t) & \text{whenever } j \in N_i^{\text{in}}(t), \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

# 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* $\lambda$ *and* $\delta$ *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. (4) 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

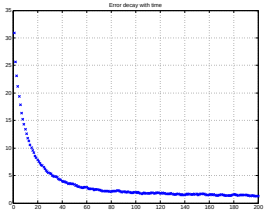
S.S. Ram, A. Nedić, and V.V. Veeravalli, "Distributed Stochastic Subgradient Projection Algorithms for Convex Optimization," Journal of Optimization Theory and Applications,147 (3) 516–545, 2010

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 $\lambda$, which is a measure of the connectivity of the directed sequence $G(1), G(2), \ldots$; namely, the closeness of $\lambda$ to 1 measures the speed at which a consensus process on the graph sequence $\{G(t)\}$ converges.

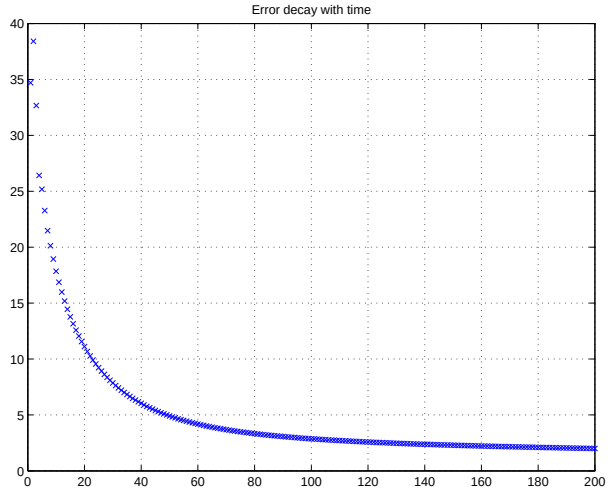
However, our bounds also include the parameter $\delta$, which, as we will later see, is a measure of the imbalance of influences among the nodes. Time-varying directed regular networks are uniform in influence and will have $\delta = 1$, so that $\delta$ will disappear from the bounds entirely; however, 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.

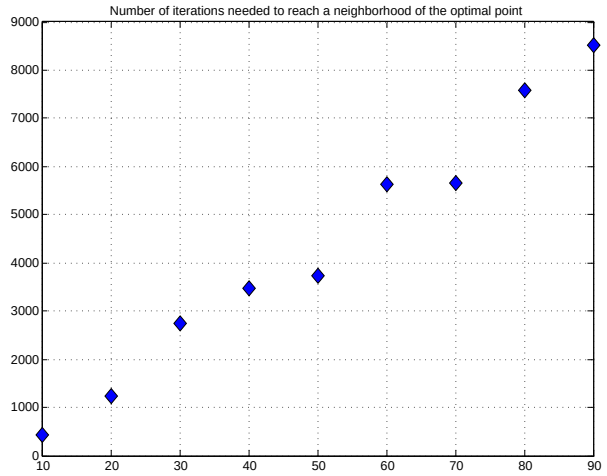
# Simulations




The details are in:

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


Error decay with time
40
35
30
25
20
15
10
5
0
0
20
40
60
80
100
120
140
160
180
200


Number of iterations needed to reach a neighborhood of the optimal point
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
10
20
30
40
50
60
70
80
90