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

joint work with Alexander Olshevsky (ISE, UIUC)
Distributed Optimization Problems: Challenges

- **Lack of central “authority”**
  - The centralized 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

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

---

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 \( x^* \)

Centralized (not distributed) formulation

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

s.t. \( (x, \xi) \in X \triangleq \{(x, \xi) \mid y_j \langle x, z_j \rangle \geq 1 - \xi_j, \xi_j \geq 0, j = 1, \ldots, p\} \)
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 $x^*$, without disclosing the data sets

\[
\min_{x \in \mathbb{R}^d, \xi \in \mathbb{R}^p} \sum_{i=1}^{m} \left( \frac{1}{2m} \|x\|^2 + C \sum_{j \in J_i} \xi_j \right) \\
\text{s.t. } (x, \xi) \in \bigcap_{i=1}^{m} X_i, \\
X_i \triangleq \{(x, \xi) \mid y_j \langle x, z_j \rangle \geq 1 - \xi_j, \xi_j \geq 0, j \in J_i\} \quad \text{for } i = 1, \ldots, m
\]
Consensus Model

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.,

$$\lim_{t \to \infty} x_i(t) = x$$

for all $i$. 

Dynamic Network Topology

Each agent dynamic is given by

\[ x_i(k + 1) = \sum_{j \in N_i(k)} a_{ij}(k)x_j(k) \]

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 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

\[ a_{ij}(k) > 0, \ j \in N_i(k) \quad \text{and} \quad \sum_{j \in N_i(k)} a_{ij}(k) = 1 \quad \text{for all } i \ \text{and} \ k \]
Weight Matrices

Introduce the weight matrix $A(k)$ which is compliant with the connectivity graph $(V, E_k)$ enlarged with the self-loops:

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

Assumption 1: For each $k$,

- The graph $(V, E_k)$ 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 $i, j, k$

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

**Proposition 2** [Tsitsiklis 84] Under Assumption 1, the agent values converge to a consensus with a geometric rate. In particular,

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

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

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

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

**The convergence rate is geometric**
Computational Model

Part II

Distributed Optimization in Network

- Optimization problem - classic
- Problem data distributed - new
General Multi-Agent Model

- Network of $m$ agents represented by an undirected graph $([m], E_t)$ where $[m] = \{1, \ldots, m\}$ and $E_t$ is the edge set

- Each agent $i$ has a \textbf{convex} objective function $f_i(x)$ known to that agent only

- Common constraint (\textbf{closed convex}) set $X$ known to all agents

The problem can be formalized:

\[
\begin{align*}
\text{minimize} & \quad F(x) \triangleq \sum_{i=1}^{m} f_i(x) \\
\text{subject to} & \quad x \in X \subseteq \mathbb{R}^n
\end{align*}
\]
How Agents Manage to Optimize Global Network Problem?

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

• 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)


Distributed Optimization Algorithm

minimize $F(x) = \sum_{i=1}^{m} f_i(x)$ 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
    
    $$w_i(t + 1) = \sum_{j=1}^{m} a_{ij}(t)x_j(t)$$
    with $a_{ij}(t) = 0$ when $j \notin N_i(t)$

  - 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":

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

\[ = w_i(t + 1) + \left(\prod_X [w_i(t + 1) - \alpha(t) \nabla f_i(w_i(t + 1))] - \prod_X [w_i(t + 1)]\right) \]

small stepsize \( \alpha(t) \)

\[ \approx w_i(t + 1) - \alpha(t) \nabla f_i(w_i(t + 1)) \]

\[ = \sum_{j=1}^{m} a_{ij}(t) x_j(t) - \alpha(t) \nabla f_i \left( \sum_{j=1}^{m} a_{ij}(t) x_j(t) \right) \]

Matrices \( A \) that lead to consensus, also yield convergence of an optimization algorithm.
Convergence Result

- Method:

\[
\begin{align*}
    w_i(t + 1) &= \sum_{j=1}^{m} a_{ij}(t) x_j(t) \\
    a_{ij}(t) &= 0 \text{ when } j \notin N_i(t) \\
    x_i(t + 1) &= \prod_X \left[ w_i(t + 1) - \alpha(t) \nabla f_i(w_i(t + 1)) \right]
\end{align*}
\]

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 \( G(t) = ([m], E_t) \) 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\(^\dagger\).

Then, for some solution \( x^* \) of the problem we have

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

Related Papers

The paper looks at a basic (sub)gradient method with a constant stepsize

The paper looks at stochastic (sub)gradient method with diminishing stepsizes and constant as well

The paper looks at extension of the method for other types of network objective functions
**Other Extensions**

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

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

Extensions include

- Gradient directions \( \nabla f_i(w_i(t + 1)) \) can be erroneous
  \[ x_i(t + 1) = \prod_X [w_i(t + 1) - \alpha(t)(\nabla f_i(w_i(t + 1)) + \varphi_i(t + 1))] \]

- 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 \( \mathcal{E}(t) \) 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 \( i \)
  \[ x_i(t + 1) = \prod_{X_i} [w_i(t + 1) - \alpha(t) \nabla f_i(w_i(t + 1))] \]
  [Nedić, Ozdaglar, and Parrilo 2010, Srivastava‡ and Nedić 2011, Lee and AN 2013]

‡Uses different weights
• Different sum-based functional structures [Ram, Nedić, and Veeravalli 2012]


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 \( x^* \)

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

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

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:

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

Distributed algorithm has the form:

\[
w_i(t + 1) = x_i(t) - \eta(t) \sum_{j=1}^{m} r_{ij} x_j(t) \quad (r_{ij} = 0 \text{ when } j \notin N_i)
\]

\[
x_i(t + 1) = w_i(t + 1) - \alpha(t) \underbrace{g_i(w_i(t + 1))}_{\text{subgradient of } f_i}
\]

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

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

$$w_i(t + 1) = x_i(t) - \eta(t) \sum_{j=1}^{m} r_{ij}(x_j(t) + \xi_{ij}(t))$$

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

$$x_i(t + 1) = w_i(t + 1) - \alpha(t) g_i(w_i(t + 1))$$

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

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

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

$$\sum_t \alpha(t) \eta(t) < \infty, \quad \sum_t \frac{\alpha^2(t)}{\eta(t)} < \infty$$
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
Advantages/Disadvantages

- 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
Push-Sum Based Computational Model

Part III

Distributed Optimization in Directed Networks

- Motivated by work of M. Rabbat, K.I. Tsianos and S. Lawlor

- The need to eliminate doubly stochastic weights and practical issues with bi-directional communications
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 $x_i(t)$ and $y_i(t)$.

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

$$x_i(t + 1) = \sum_{j \in N_{i}^{\text{in}}(t)} \frac{x_j(t)}{d_j(t)},$$

$$y_i(t + 1) = \sum_{j \in N_{i}^{\text{in}}(t)} \frac{y_j(t)}{d_j(t)},$$

$$z_i(t + 1) = \frac{x_i(t + 1)}{y_i(t + 1)}$$

(1)

- 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 $w_i(0) = z_i(0) = 1$ and $y_i(0) = 1$ for all $i$.

---


**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

\[
\left| z_i(t + 1) - \frac{1^t x(t)}{n} \right| \leq \frac{8}{\delta} \left( \lambda^t \|x(0)\|_1 + \sum_{s=1}^{t} \lambda^{t-s} \|\epsilon(s)\|_1 \right),
\]

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

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

\[ w_i(t+1) = \sum_{j \in N_i^{\text{in}}(t)} \frac{x_j(t)}{d_j(t)}, \]
\[ y_i(t+1) = \sum_{j \in N_i^{\text{in}}(t)} \frac{y_j(t)}{d_j(t)}, \]
\[ z_i(t+1) = \frac{w_i(t+1)}{y_i(t+1)}, \]
\[ x_i(t+1) = w_i(t+1) - \alpha(t+1)g_i(t+1), \quad (2) \]

where $g_i(t + 1)$ is a subgradient of the function $f_i$ at $z_i(t+1)$. The method is initiated with $w_i(0) = z_i(0) = 1$ and $y_i(0) = 1$ for all $i$. 
The stepsize $\alpha(t + 1) > 0$ satisfies the following decay conditions
\[
\sum_{t=1}^{\infty} \alpha(t) = \infty, \quad \sum_{t=1}^{\infty} \alpha^2(t) < \infty, \quad \alpha(t) \leq \alpha(s) \text{ for all } t > s \geq 1.
\] (3)

We note that the above equations have simple broadcast-based implementation: each node $i$ broadcasts the quantities $x_i(t)/d_i(t), y_i(t)/d_i(t)$ to all of the nodes in its out-neighborhood**, which simply sum all the messages they receive to obtain $w_i(t + 1)$ and $y_i(t + 1)$. The update equations for $z_i(t + 1), x_i(t + 1)$ 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

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.

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

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

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

Then, the distributed subgradient-push method of Eq. (2) with the stepsize satisfying the conditions in Eq. (3) has the following property

$$\lim_{t \to \infty} z_i(t) = z^* \quad \text{for all } i \text{ and for some } z^* \in Z^*.$$
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 $z_i(1), z_i(2), \ldots$ 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 $w_i(t+1)$ and $y_i(t+1)$ in Eq. (2), which are defined by

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

(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 \( \tilde{z}_i(t) \in \mathbb{R}^d \) initialized at time \( t = 1 \) to \( \tilde{z}_i(1) = z_i(1) \) and updated as

\[
\tilde{z}_i(t + 1) = \frac{\alpha(t + 1)z_i(t + 1) + S(t)\tilde{z}_i(t)}{S(t + 1)},
\]

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 \( z^* \in Z^* \),

\[
F(\tilde{z}_i(t)) - F(z^*) \leq \frac{n}{2} \frac{||\bar{x}(0) - z^*||_1}{\sqrt{t}} + \frac{n}{2} \left( \sum_{i=1}^{n} L_i \right)^2 \frac{(1 + \ln t)}{\sqrt{t}}
\]

\[
+ \frac{16}{\delta(1 - \lambda)} \left( \sum_{i=1}^{n} L_i \right) \frac{\sum_{j=1}^{n} ||x_j(0)||_1}{\sqrt{t}} + \frac{16}{\delta(1 - \lambda)} \left( \sum_{i=1}^{n} L_i^2 \right) \frac{(1 + \ln t)}{\sqrt{t}}
\]

where

\[
\bar{x}(0) = \frac{1}{n} \sum_{i=1}^{n} x_i(0),
\]

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

$$
\delta \geq \frac{1}{n^{nB}},
$$

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

(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


subgradient-norm upper bounds $L_i$, as well as on the initial conditions $x_i(0)$. 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

Number of iterations needed to reach a neighborhood of the optimal point