Distributed average consensus for wireless networks

Philippe Ciblat

Joint work with P. Bianchi, W. Hachem and F. Iutzeler

Institut Mines-Telecom/Telecom ParisTech, Paris, France

Outline

1. A few applications

2. Main mathematical tools
   - Graph Theory
   - Non-negative (stochastic) matrices
   - Markov Chain

3. Standard (single-variate based-) algorithms
   - Synchronous case [DeGroot1974]
   - Asynchronous case [Boyd2006, Aysal2009]

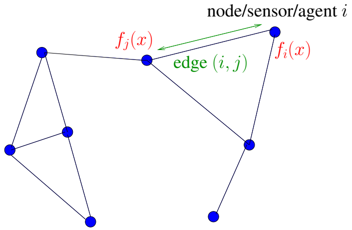
4. New (bivariate based-) algorithm

5. Numerical illustrations

6. Perspectives : distributed optimization

Part 1 : Applications

Example 1 : distributed optimization



$$\min_x f(x) \triangleq \sum_{i=1}^{N} f_i(x)$$

with

- N number of nodes
- f_i known only at node i
- No fusion center (f known nowhere)

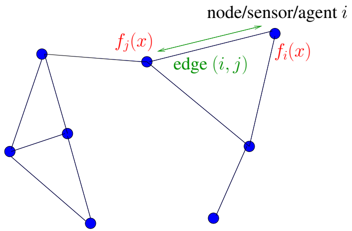
$\Rightarrow$ distributed processing

Practical functions

In sensor networks, $x = \theta$ may be the temperature (or gas pressure, ...), and f_i the log-likelihood $(y_i - \theta)^2$ where y_i is the measurement at node i. Then

$$\hat{\theta}_{\text{opt}} = \theta_{\text{ave}} = \frac{1}{N}\sum_{i=1}^{N} y_i \Leftrightarrow \textbf{average computation}$$

Example 1 : distributed optimization



$$\min_x f(x) \triangleq \sum_{i=1}^{N} f_i(x)$$

with

- N number of nodes
- f_i known only at node i
- No fusion center (f known nowhere)

$\Rightarrow$ distributed processing

Practical functions

In resource allocation, $x = [P_1, \cdots, P_N]$ may be the powers, and f_i the Shannon capacity at node i

Here, no direct average computation
but average computation will be actually needed

Example 1 : distributed optimization (cont'd)

A simple distributed average computation algorithm [DeGroot1974]

At time k, each node replaces its current value with a weighted average of its current value and those of its neighbors.

$$\mathbf{x}(k+1) = \mathbf{K}\mathbf{x}(k) \text{ with } \mathbf{K}_{i,j} = \begin{cases} \frac{1}{1+\max\{d_i, d_j\}} & \text{if } j \in \mathcal{N}_i \\ 1 - \sum_{j \in \mathcal{N}_i} \mathbf{K}_{i,j} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

where $\mathbf{x}(0) = [x_1(0), \cdots, x_N(0)]^{\mathrm{T}}$ are the initial measurements.

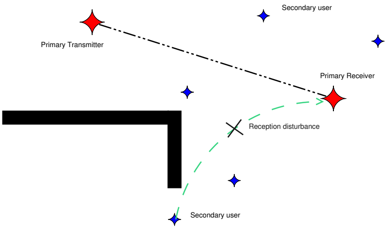
Application to distributed optimization :

Gradient step : $\tilde{x}_i(k+1) = x_i(k) - \gamma_k f_i'(x_i(k))$ for each node i.

Gossip step : $\mathbf{x}(k+1) = \mathbf{K}\tilde{\mathbf{x}}(k+1)$ with $\mathbf{K}$ used for average computation

So, average computation is needed

Example 2 : fully-distributed spectrum sensing



Problem : a secondary user is disturbing the primary receiver

Solution : secondary users have to cooperate to detect the primary user

Optimal test : Log-Likelihood Ratio (LLR)

When the transmit signal and the noise are Gaussian,

$$T = \frac{1}{N}\sum_{i=1}^{N} t_i \gtrless \eta, \quad \text{with} \quad t_i = \frac{\mathrm{SNR}_i}{\sigma_i^2(1+\mathrm{SNR}_i)}\sum_{k=1}^{N_s}|y_i(k)|^2$$

Here, average computation is needed

Part 2 : Mathematical tools

Graph model

Let $\mathcal{G} = (V, E)$ be the underlying communication graph between the nodes

- with V is the set of Vertices/nodes/sensors/agents ($N = |V|$)
- with E is the set of Edges (perfect communication links)

The communication graph $\mathcal{G}$ is assumed

- **unweighted** (each edge has the same weight equal to 1)
- **undirected** (if $i \to j$ is a link, then $j \to i$ too)
- **connected** (one can find a path from any i to any j)

Graph theory

Let $\mathcal{G}$ be an undirected but possibly weighted graph.

- $w_{i,j}$ the weight of the link (i,j)
$$\begin{cases} w_{i,j} > 0 & \text{if } (i,j) \in E \\ w_{i,j} = 0 & \text{otherwise} \end{cases}$$
- $\mathcal{N}_i = \{j \in V | (i,j) \in E\}$ the neighborhood of the node i
- $d_i = |\mathcal{N}_i|$ the degree of node i
- $\mathbf{D} = \mathrm{diag}(d_1, \cdots, d_N)$ is the degree diagonal matrix
- $\mathbf{A}$ is the adjacency matrix such that $\mathbf{A}_{i,j} = w_{i,j}$
- $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the Laplacian matrix

Remarks

- Let $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ be the eigenvalues of $\mathbf{L}$.
$$\lambda_2 \neq 0 \Leftrightarrow \mathcal{G} \text{ is connected}$$
- Let $\mathbf{S} = \mathrm{Supp}(\mathbf{A})$ s.t. $\mathbf{S}_{i,j} = 1$ if $w_{i,j} > 0$, and 0 otherwise.
As $\mathbf{S}$ adjacency matrix of the associated unweighted graph, properties on connectivity can be checked with $\mathbf{S}$ instead of $\mathbf{A}$.

Node activation model

Synchronous context :

- At time k, all nodes exchange information and do the same algebraic manipulations. So $\exists!$ $\mathbf{K}$ s.t.

$$\mathbf{x}(k+1) = \mathbf{K}\mathbf{x}(k) \quad \Rightarrow \quad \mathbf{x}(k) = \mathbf{K}^k\mathbf{x}(0)$$

Asynchronous context :

- At time k, only one node (let say i_k) exchanges information. This node and some of its neighbors do algebraic manipulations through a finite set of matrices $\mathcal{K}_{i_k} = \{\mathbf{K}_{\omega_{i_k}}\}$. So, $\exists$ $\mathcal{K} = \cup_{i=1}^{N}\mathcal{K}_i$ s.t.

$$\mathbf{x}(k+1) = \mathbf{K}_{\omega_{i_k}}\mathbf{x}(k) \;(= \mathbf{K}_{\nu_k}\mathbf{x}(k)) \quad \Rightarrow \quad \mathbf{x}(k) = \left(\prod_{\ell=1}^{k}\mathbf{K}_{\nu_\ell}\right)\mathbf{x}(0)$$

- i_k is an i.i.d process and $\mathbb{P}(i_k = j) = p_j$
- w_i is an i.i.d process $\Rightarrow$ ν_k is an i.i.d process
- Collision-free assumption (still valid for collision if CRC at RX)

Non-negative matrices

Non-Negative Matrices (NNM)

$\mathbf{M}$ is a $N \times N$ non-negative matrix iff $\mathbf{M}_{i,j} \geq 0, \quad \forall i, j$

NNM play a crucial role herafter since

- $\mathbf{A}$ the adjacency matrix of the communication graph is NNM
- $\mathbf{K}_\nu$ the update matrices are NNM as well

Actually any NNM can be viewed as an adjacency matrix of a weighted and directed graph $\mathcal{G}(\mathbf{M})$

Two types of connectivity

When the graph is directed, we can define

- weak connectivity : path from i to j OR from j to i, for any i, j
- **strong** connectivity : paths from i to j **AND** from j to i, for any i, j

Irreducible matrices

Let $\mathbf{A}$ be a $N \times N$ non-negative matrix. $\mathbf{A}$ is **irreducible** iff

– *either* $(\mathbf{I} + \mathbf{A})^{N-1} > 0$;
– *or* $\mathcal{G}(\mathbf{A})$ is strongly connected.

Perron-Frobenius theorem for irreducible matrices

Let $\rho(\mathbf{A}) \triangleq \max_i\{|\lambda_i^{\mathbf{A}}|\}$ be the *spectral radius*.

Let $\mathbf{A}$ be a $N \times N$ irreducible matrix. Then

i) $\rho(\mathbf{A}) > 0$ is a simple eigenvalue of $\mathbf{A}$;
ii) there is a positive vector $\mathbf{x}$ such that $\mathbf{Ax} = \rho(\mathbf{A})\mathbf{x}$.

Remark :

- if $\mathbf{A}$ irreducible, $\text{Supp}(\mathbf{A})$ irreducible too.

Primitive matrices

Let **A** be a $N \times N$ non-negative matrix. **A** is **primitive** iff

– $\exists\, m \geq 1$ s. t. $\mathbf{A}^m$ is a positive matrix.

Primitivity leads to irreducibility and strong connectivity of $\mathcal{G}(\mathbf{A})$

Perron-Frobenius theorem for primitive matrices

Let **A** be a $N \times N$ primitive matrix. In addition to irreducible case,

iii) $\rho(\mathbf{A})$ is the only eigenvalue of maximal modulus.

Remarks :

- If **A** irreducible but not primitive, **A** has $k > 1$ eigenvalues of maximal modulus
- If **A** irreducible with positive diagonal entries, **A** primitive
- If **A** irreducible, adding self-loop to **A** leads primitivity
- If **A** primitive, Supp(**A**) primitive too.

Stochastic matrices

Row-stochastic matrix

Non-negative matrix with sum of each row equal to 1 $\Leftrightarrow \mathbf{A1} = \mathbf{1}$

Column-stochastic matrix

Non-negative matrix with sum of each column equal to 1 $\Leftrightarrow \mathbf{1}^{\mathrm{T}}\mathbf{A} = \mathbf{1}^{\mathrm{T}}$

Doubly-stochastic matrix

Row-stochastic <u>and</u> column-stochastic matrix

If **A** is either row-stochastic or column-stochastic matrix, then

- $\rho(\mathbf{A}) = 1$
- $\mathbf{1}$ (left or right)-eigenvector for the largest eigenvalue.

Remark : K in Slide 4 is doubly-stochastic.

Link with Markov Chain

Any $N \times N$ row-stochastic matrix is a transition probability matrix of a discrete-time Markov Chain with N states and conversely.

- $\mathbf{W}_k$ transition probability matrix at time k.
- $\mathbf{t}(k)$ states distribution at time k (non-negative with $\mathbf{1}^{\mathrm{T}}\mathbf{t}(k) = 1$)

$$\mathbf{t}(k+1)^{\mathrm{T}} = \mathbf{t}(k)^{\mathrm{T}}\mathbf{W}_k$$

Remarks :

- States distribution analysis leads to right multiplication

$$\mathbf{P}_f^{s,s+k} = \mathbf{W}_s\mathbf{W}_{s+1}\cdots\mathbf{W}_{s+k} \quad \text{(forward direction)}$$

- Average computation analysis leads to left multiplication (Slide 8)

$$\mathbf{P}_b^{s,s+k} = \mathbf{W}_{s+k}\mathbf{W}_{s+k-1}\cdots\mathbf{W}_s \quad \text{(backward direction)}$$

Ergodicity of Markov Chain

Row-stochastic matrices $\{\mathbf{W}_k\}_k$ **weakly ergodic** if rows of $\mathbf{P}_f^{1,k}$ tend to be identical, but may vary. $\exists$ $\mathbf{v}(k)$ non-negative vector ($\mathbf{1}^{\mathrm{T}}\mathbf{v}(k) = 1$),

$$\mathbf{P}_f^{1,k} \overset{k\to\infty}{\sim} \mathbf{1}\mathbf{v}(k)^{\mathrm{T}}$$

Application to our problem : nodes agree but agreement changes.

Row-stochastic matrices $\{\mathbf{W}_k\}_k$ **strongly ergodic** if rows of $\mathbf{P}_f^{1,k}$ tend to be identical. $\exists$ $\mathbf{v}$ non-negative vector ($\mathbf{1}^{\mathrm{T}}\mathbf{v} = 1$),

$$\mathbf{P}_f^{1,k} \xrightarrow{k\to\infty} \mathbf{1}\mathbf{v}^{\mathrm{T}}$$

Application to our problem : nodes reach a consensus ($= \mathbf{v}^{\mathrm{T}}\mathbf{x}(0)$).

Remarks :

- If $\{\mathbf{W}_k\}_k$ column-stochastic (not row-), replacing $\mathbf{W}_k$ with $\mathbf{W}_k^{\mathrm{T}}$
- If $\{\mathbf{W}_k\}_k$ doubly-stochastic, $\mathbf{v} = (1/N)\mathbf{1}$
- If backward considered, same definition with $\mathbf{P}_b^{1,k}$ instead of $\mathbf{P}_f^{1,k}$. Here, weak ergodicity=strong ergodicity [Chatterjee1977]

Part 3 : Standard (single-variate based-) algorithms

Model

$$\mathbf{x}(k+1) = \mathbf{K}_{\xi_{k+1}} \mathbf{x}(k)$$

- $\mathbf{x}(k) = [x_1(k), \cdots, x_N(k)]^{\mathrm{T}}$ where $x_i(k)$ value of node i at time k
- $\{\mathbf{K}_{\xi_k}\}_k$ i.i.d. process valued in $\mathcal{K} = \{\mathbf{K}_j\}_{j=1,\dots,M}$
- $\mathbf{K}_{\xi_k}$ non-negative, positive diagonal, and support included in that of $\mathbf{I} + \mathbf{A}$ with $\mathbf{A}$ communication graph adjacency matrix

Sum conservation : average of the initial values kept at any time

$$\mathbf{1}^{\mathrm{T}}\mathbf{x}(k+1) = \mathbf{1}^{\mathrm{T}}\mathbf{x}(k) \;\Leftrightarrow\; \textbf{column-stochastic}$$

Consensus conservation : consensus stable (if it exists)

$$\mathbf{x}(k) = c\mathbf{1}, \text{so } \mathbf{x}(k+1) = c\mathbf{1} \;\Leftrightarrow\; \textbf{row-stochastic}$$

Let $\mathbf{J}$ be the projection on span$(\mathbf{1})$: $\mathbf{J} = (1/N)\mathbf{1}\mathbf{1}^{\mathrm{T}}$
Let $\mathbf{J}^{\perp}$ be the projection on span$(\mathbf{1})^{\perp}$: $\mathbf{J}^{\perp} = \mathbf{I} - \mathbf{J}$.

Doubly-stochastic matrices - sync. case

$$\mathbf{x}(k+1) = \mathbf{K}\mathbf{x}(k) \quad \text{with } \mathbf{K} \text{ doubly-stochastic}$$

- $\mathbf{K}$ row-stochastic, so transition probability matrix of homogeneous Markov Chain (weak=strong, forward=backward)
- If $\mathbf{K}$ primitive, Markov Chain ergodic : $\lim_{m\to\infty} \mathbf{K}^m = \mathbf{1}\mathbf{v}^{\mathrm{T}}$
- $\mathbf{K}$ column-stochastic, so $\mathbf{v} = (1/N)\mathbf{1}$.

Then

$$\mathbf{x}(k) \to x_{\text{ave}}\mathbf{1} \quad \left(\text{with } x_{\text{ave}} = \frac{1}{N}\sum_{i=1}^{N} x_i(0)\right)$$

Primitivity ?

$\mathbf{K}$ primitive if

- communication graph connected
- support of $\mathbf{K}$ identical to that of communication graph (+self-loop)

Example : Metropolis algorithm (Slide 4)

Doubly-stochastic matrices - sync. case (cont'd)

What is the convergence speed ?

- If $\mathbf{K}$ primitive, $\mathbf{K}\mathbf{K}^{\mathrm{T}}$ primitive. As doubly-stochastic,

$$\rho(\mathbf{K}\mathbf{K}^{\mathrm{T}} - \mathbf{J}) < 1$$

- Why $\rho(\mathbf{K}\mathbf{K}^{\mathrm{T}} - \mathbf{J})$ plays a great role :

$$\mathbf{x}(k) = \mathbf{J}\mathbf{x}(k) + \mathbf{J}^{\perp}\mathbf{x}(k) \overset{\mathbf{K}\ col.-sto.}{=} x_{\mathrm{ave}}\mathbf{1} + \mathbf{J}^{\perp}\mathbf{x}(k)$$

Convergence of $\mathbf{x}(k) \sim$ convergence of $\|\mathbf{J}^{\perp}\mathbf{x}(k)\|_2^2$

- In addition

$$\|\mathbf{J}^{\perp}\mathbf{x}(k+1)\|_2^2 \overset{\mathbf{K}\ row.-sto.}{\leq} \rho\left(\mathbf{K}\mathbf{K}^{\mathrm{T}} - \mathbf{J}\right) \|\mathbf{J}^{\perp}\mathbf{x}(k)\|_2^2.$$

Link between projections into $\mathrm{span}(\mathbf{1})^{\perp}$ at time k and $(k+1)$

Main result

Under mild assumptions, doubly-stochastic update matrices lead to linear convergence with slope $\alpha = \rho(\mathbf{K}\mathbf{K}^{\mathrm{T}} - \mathbf{J}) < 1$

Doubly-stochastic matrices - async. case

$$\mathbf{x}(k) = \mathbf{P}^{1,k}\mathbf{x}(0) \quad \text{with} \quad \mathbf{P}^{1,k} \triangleq \mathbf{K}_{\xi_k}\mathbf{K}_{\xi_{k-1}}\ldots\mathbf{K}_{\xi_1}.$$

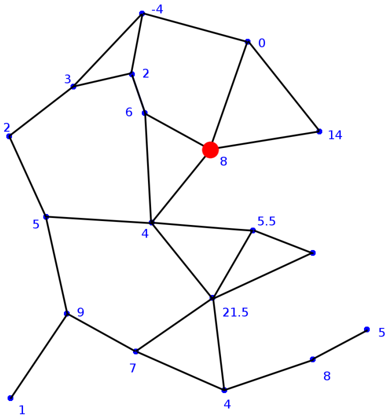
with $\{\mathbf{K}_{\xi_k}\}_k$ assumed i.i.d..

- As $\mathbf{K}_i$ row-stochastic, $\mathbf{P}^{1,k}$ **backward** concatenation of transition probability matrix of a heterogeneous Markov Chain
- Using similar arguments as previous slide, (strong/weak) ergodicity is needed and ensured by primitivity of $\mathbb{E}[\mathbf{K}]$.
- $\mathbb{E}[\mathbf{K}]$ primitive if communication graph connected and support of $\mathbb{E}[\mathbf{K}]$ identical to that of communication graph (+self-loop)

Main result

Under mild assumptions, doubly-stochastic update matrices lead to linear convergence with slope $\beta = \rho(\mathbb{E}[\mathbf{K}\mathbf{K}^{\mathrm{T}}] - \mathbf{J}) < 1$

Example : Random Pairwise Gossip [Boyd2006]

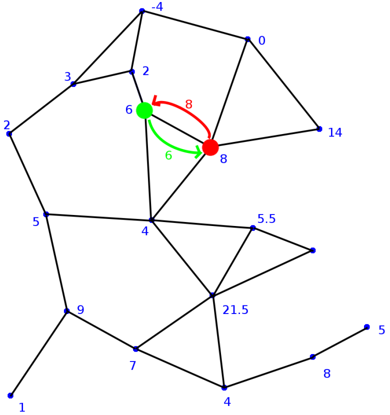


At time k, let i be the active node

- i chooses a neighbor j uniformly in $\mathcal{N}_i$
- i and j exchange their values
- i and j update : $x_i(k+1) = x_j(k+1) = \frac{x_i(k)+x_j(k)}{2}$.

All results given in previous slide hold

Example : Random Pairwise Gossip [Boyd2006]

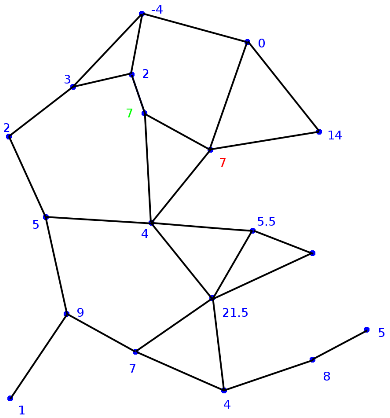


At time k, let i be the active node

- i chooses a neighbor j uniformly in $\mathcal{N}_i$
- i and j exchange their values
- i and j update : $x_i(k+1) = x_j(k+1) = \frac{x_i(k)+x_j(k)}{2}$.

All results given in previous slide hold

Example : Random Pairwise Gossip [Boyd2006]



At time k, let i be the active node

- i chooses a neighbor j uniformly in $\mathcal{N}_i$
- i and j exchange their values
- i and j update : $x_i(k+1) = x_j(k+1) = \frac{x_i(k)+x_j(k)}{2}$.

All results given in previous slide hold

Example : Random Pairwise Gossip [Boyd2006]

$$\mathbf{K}_{\{i,j\}} = \begin{bmatrix} \ddots 1 \ddots & & & & \\ & 1/2 & & 1/2 & \\ & & \ddots 1 \ddots & & \\ & 1/2 & & 1/2 & \\ & & & & \ddots 1 \ddots \end{bmatrix}$$

At time k, let i be the active node

- i chooses a neighbor j uniformly in $\mathcal{N}_i$
- i and j exchange their values
- i and j update : $x_i(k+1) = x_j(k+1) = \frac{x_i(k)+x_j(k)}{2}$.

All results given in previous slide hold

Non-doubly stochastic matrices

Doubly-stochastic update matrices require FEEDBACK

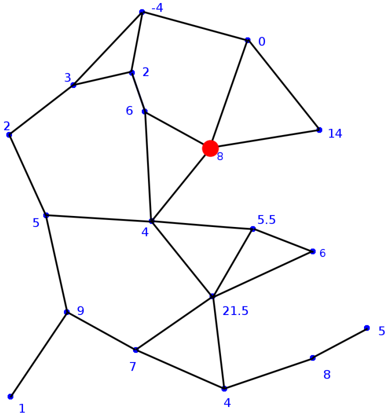
- Feedback (especially if routing) needs slow-varying network
- Feedback prevents to take benefit of **broadcast nature of wireless channel**

Two cases : Row-stochastic or Column-stochastic

Row-stochastic case

- **Design :** easy by doing weighted mean at RX
- **Problem :** no sum conservation ; information is lost (Slide 16) ! ! !
- **Result :** strong ergodicity of Markov Chain (Slide 14)
  - If $\mathbb{E}[\mathbf{K}]$ primitive, $\mathbf{x}(k) \stackrel{a.s.}{\to} (\mathbf{v}^{\mathrm{T}}\mathbf{x}(0))\mathbf{1}$
  - In addition, if $\mathbb{E}[\mathbf{K}]$ doubly-stochastic, $\mathbb{E}[\mathbf{v}^{\mathrm{T}}\mathbf{x}(0)] = x_{\mathrm{ave}}$
  - Unbiased but not consistent [Tahbaz2008]
- **Conclusion :** avoid to do that ! ! !

Example : Broadcast Gossip [Aysal2009]

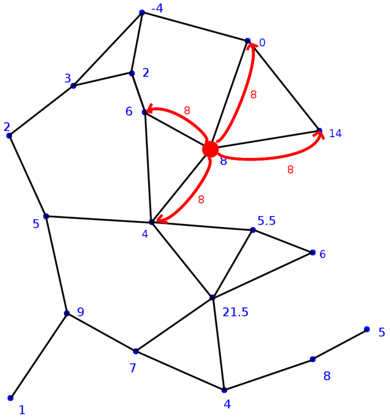


At time k, let i be the active node

- i broadcasts its value to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates : $x_j(k+1) = \frac{x_i(k)+x_j(k)}{2}$.

- Take into account broadcast nature of wireless channel
- No column-stochastic update matrices
- Do not converge to the true value (only in expectation)

Example : Broadcast Gossip [Aysal2009]

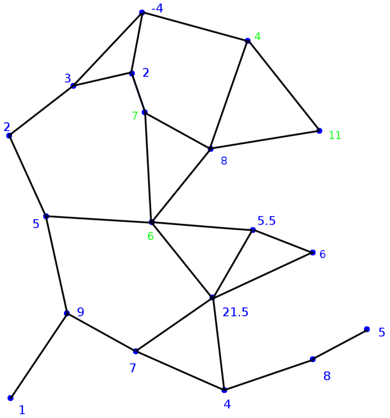


At time k, let i be the active node

- i broadcasts its value to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates :
  $x_j(k+1) = \frac{x_i(k)+x_j(k)}{2}$.

- Take into account broadcast nature of wireless channel
- No column-stochastic update matrices
- Do not converge to the true value (only in expectation)

Example : Broadcast Gossip [Aysal2009]



At time k, let i be the active node

- i broadcasts its value to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates :
  $x_j(k+1) = \frac{x_i(k)+x_j(k)}{2}$.

- Take into account broadcast nature of wireless channel
- No column-stochastic update matrices
- Do not converge to the true value (only in expectation)

Example : Broadcast Gossip [Aysal2009]

$$\mathbf{K}_i = \begin{bmatrix} 1/2 & & 1/2 & & \\ & \ddots & & & \\ & & 1 & & \\ & & 1/2 & 1/2 & \\ & & & & \ddots \end{bmatrix}$$

At time k, let i be the active node

- i broadcasts its value to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates : $x_j(k+1) = \frac{x_i(k)+x_j(k)}{2}$.

- Take into account broadcast nature of wireless channel
- No column-stochastic update matrices
- Do not converge to the true value (only in expectation)

Column-stochastic case

It seems worse ! ! !

- As column-sto., no transition probability matrices, but work on

$$(\mathbf{K}_{\xi_k} \cdots \mathbf{K}_{\xi_1})^{\mathrm{T}} = \mathbf{K}_{\xi_1}^{\mathrm{T}} \cdots \mathbf{K}_{\xi_k}^{\mathrm{T}} \text{ instead on } \mathbf{K}_{\xi_k} \cdots \mathbf{K}_{\xi_1}$$

Row-stochastic “update matrices “ in FORWARD direction

- In forward direction, one can only ensure weak ergodicity.
  $\exists$ $\mathbf{v}(k)$ non-negative vectors (with $\mathbf{1}^{\mathrm{T}}\mathbf{v}(k) = 1$) s.t.

$$(\mathbf{K}_{\xi_k} \cdots \mathbf{K}_{\xi_1})^{\mathrm{T}} \sim \mathbf{1}\mathbf{v}(k)^{\mathrm{T}} \;\Leftrightarrow\; \mathbf{K}_{\xi_k} \cdots \mathbf{K}_{\xi_1} \sim \mathbf{v}(k)\mathbf{1}^{\mathrm{T}}.$$

Three fundamental differences :

- **no consensus anymore** as $\mathbf{x}(k) \sim (Nx_{\mathrm{ave}})\mathbf{v}(k)$, so $x_i(k) \propto v_i(k)$
- **average is available somewhere since $x_i(k) \propto x_{\mathrm{ave}}$ but hidden**
- **$\mathbf{v}(k)$ multiplies 1 by the left (not by the right as in row-sto.)**

- sum conservation : information is available
- **How recovering it ? How removing $v_i(k)$? $\Rightarrow$ side information**

Part 4 : New (bivariate based-) algorithm

Algorithm principle

An other variable providing $v_i(k)$ has to be computed in parallel

Let $\mathbf{w}(k)$ be this additional variable. We have easily

$$\mathbf{w}(k) \sim N\mathbf{v}(k), \quad \text{if} \quad \mathbf{w}(0) = \mathbf{1} \text{ and } \mathbf{w}(k+1) = \underbrace{\mathbf{K}_{\xi_{k+1}}}_{\text{same as for first variable}} \mathbf{w}(k)$$

Sum-Weight algorithms [Kempe2003, Benezit2010, Iutzeler2013a]

Let $\mathbf{s}(0) = \mathbf{x}(0)$ (sum) and $\mathbf{w}(0) = \mathbf{1}$ (weight). At time k, we have

$$\begin{aligned}
\mathbf{s}(k+1) &= \mathbf{K}_{\xi_{k+1}} \mathbf{s}(k) \\
\mathbf{w}(k+1) &= \mathbf{K}_{\xi_{k+1}} \mathbf{w}(k) \\
\mathbf{x}(k+1) &= \mathbf{s}(k+1) \oslash \mathbf{w}(k+1) \Leftrightarrow x_i(k+1) = \frac{s_i(k+1)}{w_i(k+1)}
\end{aligned}$$

Remarks :

- [Kempe2003] convergence speed for one special algorithm
- [Benezit2010] convergence proof, no convergence speed

Main contributions

Our results [Iutzeler2013a]

- Convergence speed for any Sum-Weight algorithm
- Sum-Weight algorithm using broadcast nature of wireless channel

Assumption : we remind $\mathbf{K}_\xi$ column-stochastic.

- As update matrices are not row-stochastic, recursion on $\mathbf{J}^\perp\mathbf{x}(k)$ does not work anymore
- Actually,

$$\mathbf{x}(k) = x_{\mathrm{ave}}\mathbf{1} + \mathbf{P}^{1,k}\mathbf{J}^\perp\mathbf{x}(0) \oslash \mathbf{w}(k)$$

$$\text{Convergence} \;\Leftrightarrow\; \|\mathbf{P}^{1,k}\mathbf{J}^\perp\mathbf{x}(0) \oslash \mathbf{w}(k)\|_2^2 \to 0$$

Steps for proof

$$\|\mathbf{x}(k) - x_{\text{ave}}\mathbf{1}\|_2^2 \leq \Psi_1(k)\Psi_2(k)$$

with

$$\begin{aligned}\Psi_1(k) &= \frac{\|\mathbf{x}(0)\|_2^2}{[\min_i w_i(k)]^2} \\ \Psi_2(k) &= \left\|\mathbf{P}^{1,k}\mathbf{J}^{\perp}\right\|_{\text{Frobenius}}^2\end{aligned}$$

Two main steps :

- proving that $\mathbf{w}(k)$ not often close to 0 through asymptotic analysis of $\Psi_1(k)$
- proving that $\mathbf{P}^{1,k}\mathbf{J}^{\perp}\mathbf{x}(0)$ vanishes for any $\mathbf{x}(0)$ through asymptotic analysis of $\Psi_2(k)$
- Approach inspired by [Kempe2003] but many differences

Results on $\Psi_1(k)$

Theorem

Let L be a well chosen positive constant. There exists a increasing sequence $(\to \infty)$ τ_n s.t.

$$\Psi_1(\tau_n) \leq \frac{\|\mathbf{x}(0)\|_2^2}{(m_{\mathcal{K}})^{2L}}$$

where

- $m_{\mathcal{K}}$ is the smallest non-null terms of matrices in $\mathcal{K}$,
- $\Delta_n = \tau_n - \tau_{n-1}$ is positive i.i.d. geometrically distributed random variables.

Remarks :

- $(\tau_n)_{n>0}$ are all finite and converge to infinity almost surely.
- So there is a constant $C < \infty$ s. t. the event $\{\Psi_1(k) \leq C\}$ occurs infinitely often almost surely.

Results on $\Psi_2(k)$

First naive approach [Kempe2003] :

$$\mathbb{E}[\Psi_2(k)] = \mathrm{Trace}\left((\mathbf{I}-\mathbf{J})\mathbb{E}\left[\mathbf{P}^{1,k}(\mathbf{P}^{1,k})^{\mathrm{T}}\right](\mathbf{I}-\mathbf{J})\right)$$

Then

$$\mathbb{E}[\Psi_2(k)] \leq \rho(\mathbf{M})\Psi_2(k-1), \text{ with } \mathbf{M} = \mathbf{J}^{\perp}\mathbb{E}\left[\mathbf{K}^{\mathrm{T}}\mathbf{K}\right]\mathbf{J}^{\perp}.$$

Unfortunately, $\rho(\mathbf{M}) > 1$ for some algorithms and communication graphs (especially ours -cf. Slide 29-), while the algorithm converges.

Previous equation not tight enough

Second promising approach [Iutzeler2013a] :

$$\Psi_2(k) = \|\mathbf{\Theta}(k)\|_{\mathrm{Frobenius}}, \text{ with } \mathbf{\Theta}(k) = \mathbf{P}^{1,k}\mathbf{J}^{\perp} \otimes \mathbf{P}^{1,k}\mathbf{J}^{\perp}$$

$$\mathbb{E}[\mathbf{\Theta}(k)] \leq \rho(\mathbf{R})\mathbf{\Theta}(k-1), \text{ with } \mathbf{R} = \mathbb{E}[\mathbf{K}\otimes\mathbf{K}]\ (\mathbf{J}^{\perp}\otimes\mathbf{J}^{\perp}).$$

Results on $\Psi_2(k)$ (cont'd)

Theorem

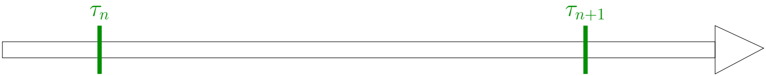
Under mild assumptions, one can prove that

$$\rho(\mathbf{R}) < 1$$

and, that

$$\mathbb{E}[\Psi_2(k)] = \mathcal{O}\left(e^{-\delta k}\right)$$

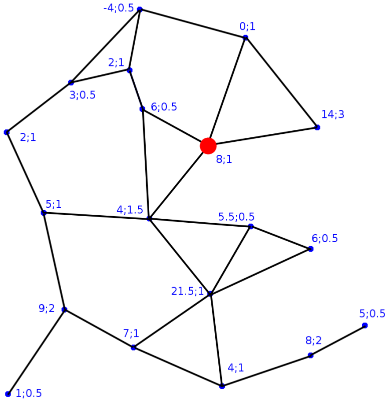
with $\delta = -\log\left(\rho\left(\mathbf{R}\right)\right) > 0$



$\|\mathbf{x}(\tau_n) - x_{\text{ave}}\mathbf{1}\|_2^2 < Ce^{-w\tau_n}$ $\|\mathbf{x}(\tau_{n+1}) - x_{\text{ave}}\mathbf{1}\|_2^2 < Ce^{-w\tau_{n+1}}$

$\max |x_i(k) - x_{\text{ave}}|^2$ is non-increasing

Example : Broadcast sum-Weight (BW) Gossip

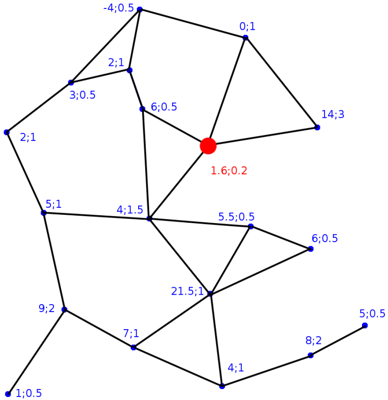


At time k, let i be the activate node [Iutzeler2013a]

- Node i updates :
$$\begin{cases} s_i(k+1) = \frac{s_i(k)}{d_i+1} \\ w_i(k+1) = \frac{w_i(k)}{d_i+1} \end{cases}$$
- i broadcasts $\left(\frac{s_i(k)}{d_i+1}; \frac{w_i(k)}{d_i+1}\right)$ to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates :
$$\begin{cases} s_j(k+1) = s_j(k) + \frac{s_i(k)}{d_i+1} \\ w_j(k+1) = w_j(k) + \frac{w_i(k)}{d_i+1} \end{cases}$$

- Take into account broadcast nature of wireless channel
- No row-stochastic matrices but converges to the true value

Example : Broadcast sum-Weight (BW) Gossip

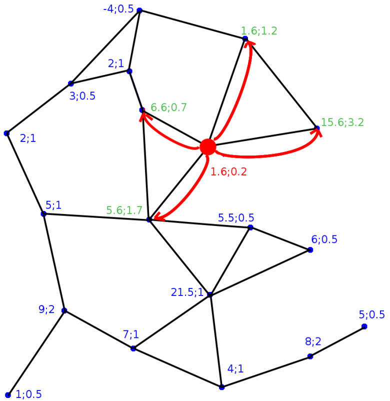


At time k, let i be the activate node [Iutzeler2013a]

- Node i updates :
$$\begin{cases} s_i(k+1) = \frac{s_i(k)}{d_i+1} \\ w_i(k+1) = \frac{w_i(k)}{d_i+1} \end{cases}$$
- i broadcasts $\left(\frac{s_i(k)}{d_i+1}; \frac{w_i(k)}{d_i+1}\right)$ to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates :
$$\begin{cases} s_j(k+1) = s_j(k) + \frac{s_i(k)}{d_i+1} \\ w_j(k+1) = w_j(k) + \frac{w_i(k)}{d_i+1} \end{cases}$$

- Take into account broadcast nature of wireless channel
- No row-stochastic matrices but converges to the true value

Example : Broadcast sum-Weight (BW) Gossip

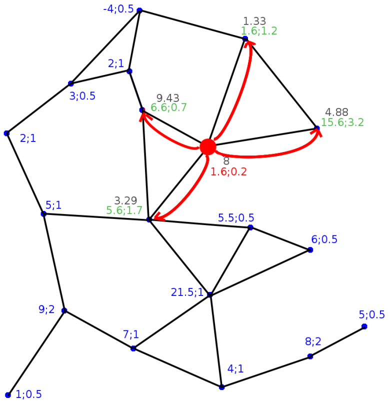


At time k, let i be the activate node [Iutzeler2013a]

- Node i updates :
$$\begin{cases} s_i(k+1) = \frac{s_i(k)}{d_i+1} \\ w_i(k+1) = \frac{w_i(k)}{d_i+1} \end{cases}$$
- i broadcasts $\left(\frac{s_i(k)}{d_i+1}; \frac{w_i(k)}{d_i+1}\right)$ to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates :
$$\begin{cases} s_j(k+1) = s_j(k) + \frac{s_i(k)}{d_i+1} \\ w_j(k+1) = w_j(k) + \frac{w_i(k)}{d_i+1} \end{cases}$$

- Take into account broadcast nature of wireless channel
- No row-stochastic matrices but converges to the true value

Example : Broadcast sum-Weight (BW) Gossip



At time k, let i be the activate node [Iutzeler2013a]

- Node i updates :
$$\begin{cases} s_i(k+1) = \frac{s_i(k)}{d_i+1} \\ w_i(k+1) = \frac{w_i(k)}{d_i+1} \end{cases}$$
- i broadcasts $\left(\frac{s_i(k)}{d_i+1}; \frac{w_i(k)}{d_i+1}\right)$ to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates :
$$\begin{cases} s_j(k+1) = s_j(k) + \frac{s_i(k)}{d_i+1} \\ w_j(k+1) = w_j(k) + \frac{w_i(k)}{d_i+1} \end{cases}$$

- Take into account broadcast nature of wireless channel
- No row-stochastic matrices but converges to the true value

Example : Broadcast sum-Weight (BW) Gossip

$$\mathbf{K}_i = \begin{bmatrix} \ddots 1 \ddots & & \frac{1}{d_i+1} & \\ & \ddots 1 \ddots & & \\ & & \frac{1}{d_i+1} & \\ & & & \ddots 1 \ddots \end{bmatrix}$$

At time k, let i be the activate node [Iutzeler2013a]

- Node i updates :
$$\begin{cases} s_i(k+1) = \frac{s_i(k)}{d_i+1} \\ w_i(k+1) = \frac{w_i(k)}{d_i+1} \end{cases}$$
- i broadcasts $\left(\frac{s_i(k)}{d_i+1}; \frac{w_i(k)}{d_i+1}\right)$ to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates :
$$\begin{cases} s_j(k+1) = s_j(k) + \frac{s_i(k)}{d_i+1} \\ w_j(k+1) = w_j(k) + \frac{w_i(k)}{d_i+1} \end{cases}$$

- Take into account broadcast nature of wireless channel
- No row-stochastic matrices but converges to the true value

Back to [Boyd2006]

Remark

If matrices in $\mathcal{K}$ are doubly-stochastic, then

$$w_i(k) = 1 \quad \Leftrightarrow \quad \Psi_1(k) = 1$$

and

$$s_i(k) = x_i(k), \quad \forall i, k$$

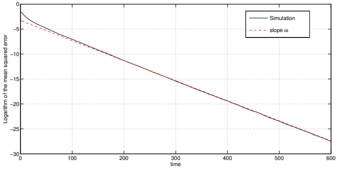
Our analysis still holds for standard single-variate algorithms

- In [Boyd2006], linear convergence with $\rho(\mathbb{E}[\mathbf{K}\mathbf{K}^{\mathrm{T}}] - \mathbf{J})$
- In [Iutzeler2013a], linear convergence with $\rho(\mathbb{E}[\mathbf{K} \otimes \mathbf{K}](\mathbf{J}^{\perp} \otimes \mathbf{J}^{\perp})$

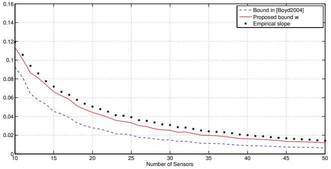
Part 5 : Numerical illustrations

Tightness of the analysis

- Random Geometric Graphs with radius of order $\sqrt{\log(N)/N}$
- Unless otherwise staded, $N = 100$



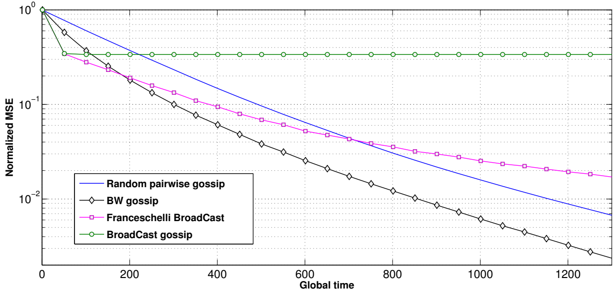
MSE for BW Gossip vs. time



Slope for Random Gossip vs. N

The proposed bound is very tight

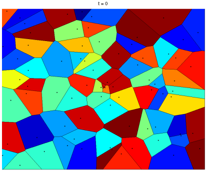
Comparison with existing methods



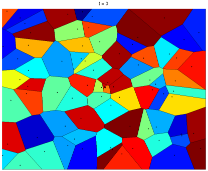
MSE for various averaging algorithms

Proposed algorithm outperforms existing ones

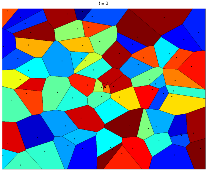
A funny example



RANDOM GOSSIP



BROADCAST GOSSIP



BW GOSSIP

Min Max Mean

Part 6 : Perspectives

Sum-Weight based distributed optimization

Go back to Slide 4 : combining Sum-Weight to gradient algorithm

1. $\tilde{\mathbf{x}}(k+1) = \mathbf{x}(k) - \gamma_k \nabla f(\mathbf{x}(k))$
2. $\mathbf{x}(k+1) = \mathbf{K}\tilde{\mathbf{x}}(k+1)$

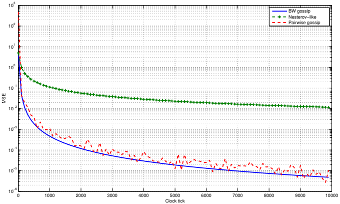
Sum-Weight based distributed optimization

Go back to Slide 4 : combining Sum-Weight to gradient algorithm

1.
$$\begin{aligned} \tilde{\mathbf{s}}(k+1) &= \mathbf{s}(k) - \gamma_k \mathbf{w}(k) \\ &\odot \nabla f(\mathbf{s}(k) \oslash \mathbf{w}(k)) \end{aligned}$$

with $\odot$ the Hadamard product

2.
$$\begin{cases} \mathbf{s}(k+1) = \mathbf{K}\tilde{\mathbf{s}}(k+1) \\ \mathbf{w}(k+1) = \mathbf{K}\mathbf{w}(k) \end{cases}$$



MSE vs time (synchronous case)

Extension to asynchronous case to be done (instability issue)

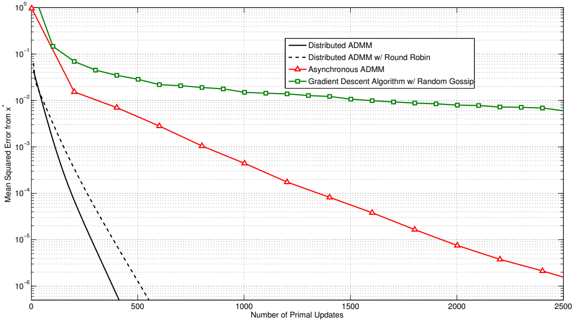
ADMM based distributed optimization

Instead improving gossip step, let us improve gradient step

$\Rightarrow$ ADMM (Alternating Direction Method of Multipliers

[Schizas2008] for synchronous case

[Iutzeler2013b] for asynchronous case



MSE vs time (asynchronous case)

Convergence speed to be evaluated ?

References

[DeGroot1974] M. DeGroot, “Reaching a consensus,” Jnl of ASA, 1974.
[Chatterjee1977] S. Chatterjee et al., “Towards consensus : some convergence theorems on repeating averaging,” Jnl of Applied Proba, 1977.
[Kempe2003] D. Kempe et al., “Gossip-based computation of aggregate infomration,” IEEE FOCS, 2003.
[Boyd2006] S. Boyd et al., “Randomized Gossip Algorithms,” IEEE TIT, 2006.
[Schizas2008] I. Schizas et al., “Consensus in ad hoc WSN with noisy links : distributed estimation of deterministic signals,” IEEE TSP, 2008.
[Tahbaz2008] A. Tahbza-Salehi et al., “A necessary and sufficient condition for consensus over random networks,” IEEE TAC, 2008.
[Aysal2009] T.C. Aysal et al., “Broadcast Gossip Algorithms for Consensus,” IEEE TSP, 2009.
[Benezit2010] F. Bénézit et al., “Weighted Gossip : distributed averaging using non-doubly-stochastic matrices ,” IEEE ISIT, 2010.
[Iutzeler2013a] F. Iutzeler et al., “Analysis of Sum-Weight-like algorithms for averaging in WSN,” IEEE TSP, 2013.
[Iutzeler2013b] F. Iutzeler et al., “Asynchronous Distributed Optimization using a Randomized ADMM,” IEEE CDC, 2013.