Gossip algorithms: Tutorial & Recent advances

Part I: Gossiping

Franck Iutzeler LJK, Université Grenoble Alpes

Smile Paris – November 3, 2016

- History & Motivation
- Mathematical formulation
- Pb. 1: Rumor Spreading
- Pb. 2: Voting
- Pb. 3: Average Consensus
  - the Synchronous Case
  - the Asynchronous Case
  - Communications and averaging

- History & Motivation

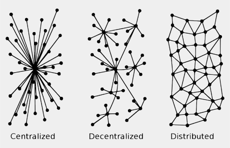
Mathematical formulation

Pb. 1: Rumor Spreading

Pb. 2: Voting

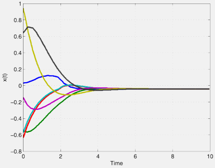
Pb. 3: Average Consensus
  - the Synchronous Case
  - the Asynchronous Case
  - Communications and averaging

Gossipping is *spreading a rumor* by *local chats*

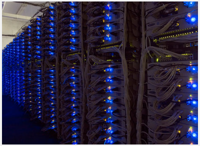


It is generally not *centralized*

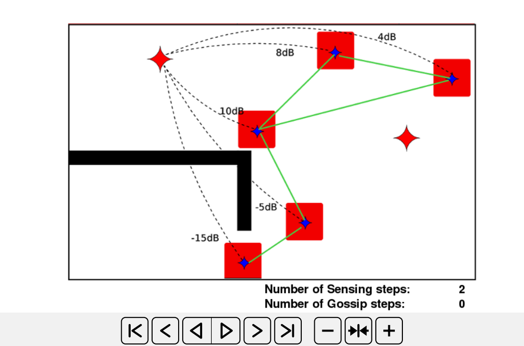
The goal is to reach a *consensus*
Often, it is not *synchronized*



- DeGroot *Reaching a consensus*
  Journal of the American Statistical Association, 1974
  Consensus as the result of a Markov chain
- Tsitsiklis & Bertsekas *Distributed asynchronous optimal routing in data networks*
  IEEE Transactions on Automatic Control, 1986
  Decentralized gradient algorithm with mixing matrices
- Demers et al. *Epidemic algorithms for replicated database maintenance*
  ACM Principles of distributed computing & Xerox note, 1987
  Epidemic algorithms for database maintenance
- Boyd et al. *Randomized gossip algorithms*
  IEEE/ACM Transactions on Networking, 2006
  Randomized Averaged Gossip



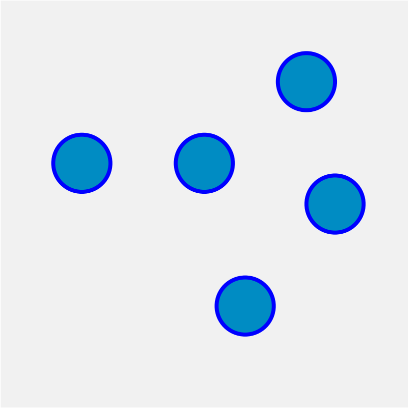
- Distributed computation, estimation
- Large-Scale optimization
- Energy production
- Fleet of UAVs, Drones, ...
- Cognitive radio



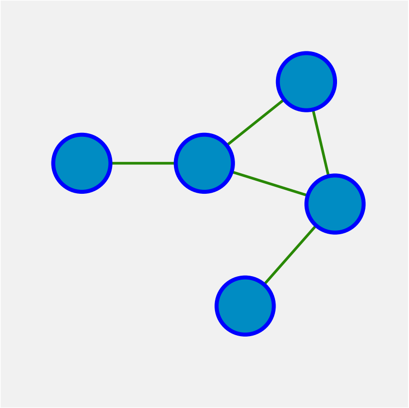
Individual hypothesis testing then Gossip:
One warns others, Majority vote, Figure of merit?

- History & Motivation
- Mathematical formulation
- Pb. 1: Rumor Spreading
- Pb. 2: Voting
- Pb. 3: Average Consensus
  - the Synchronous Case
  - the Asynchronous Case
  - Communications and averaging

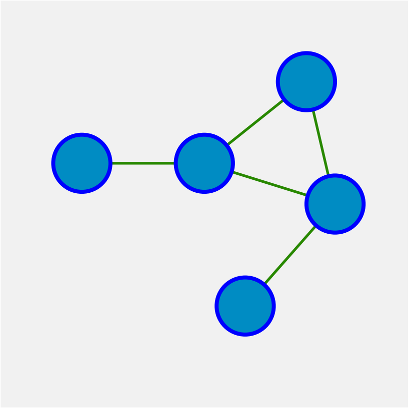
- DISTRIBUTED COMPUTATION
  estimation/optimization



- DISTRIBUTED COMPUTATION
  estimation/optimization
- AGENTS a.k.a. nodes
  local data, measurements

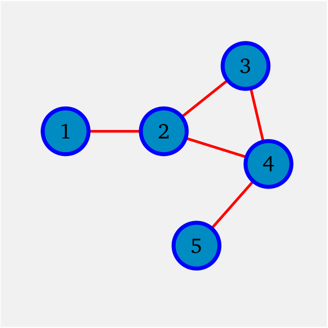


- DISTRIBUTED COMPUTATION
  estimation/optimization
- AGENTS a.k.a. nodes
  local data, measurements
- NETWORK
  wired or wireless links



- DISTRIBUTED COMPUTATION
  estimation/optimization
- AGENTS a.k.a. nodes
  local data, measurements
- NETWORK
  wired or wireless links

Goal: *distributively* reach a *consensus* over a *global value* of interest



We define a graph as $\mathcal{G} = (V, E)$

- V : set of $N = |V|$ vertices/nodes/agents
  $i \in V = \{1, 2, 3, 4, 5\}$ is a node
- E : set of edges/links
  $(i, j) \in E = \{(1, 2), (2, 3), \ldots\}$ is an edge

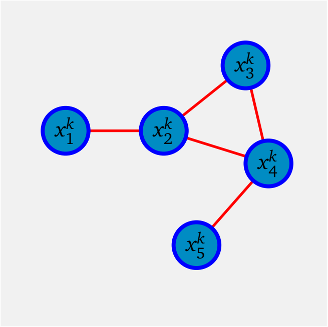
We denote:

- $\mathcal{N}_i = \{j : (i, j) \in E\}$: neighbors of i
- $d_i = |\mathcal{N}_i|$: degree of i
- A : adjacency matrix
  $N \times N$ matrix s.t. $A_{i,j} = 1$ iff i and j are neighbors

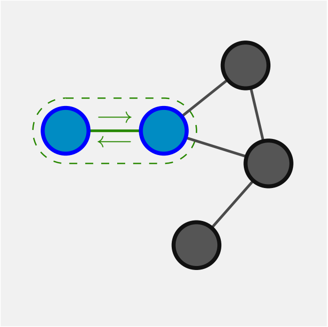
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

- L : Laplacian matrix
  $N \times N$ matrix s.t. $L = D - A$

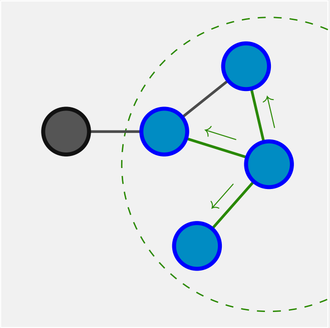
$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$



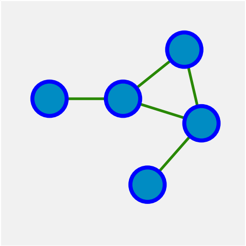
- AGENTS
  At each (discrete) time k:
    Sensor i has estimate x_i^k
    One or more perform an action
- COMMUNICATIONS



- AGENTS
  At each (discrete) time k:
  - Sensor i has estimate x_i^k
  - One or more perform an action
- COMMUNICATIONS
  - Pairwise communications



- AGENTS
  At each (discrete) time k:
    - Sensor i has estimate x_i^k
    - One or more perform an action
- COMMUNICATIONS
    - Pairwise communications
    - Broadcast communications



Goal: *distributively* reach a *consensus* over a *global value* of interest

- $x^k = (x_1^k, .., x_n^k) \to (f(x^0), .., f(x^0)) = \mathbf{1} f(x^0)$
- $x_i^{k+1} = \mathcal{F}(x_i^k, \{x_j^k : j \in \mathcal{N}_i^k\})$

Problem 1

Spreading a **rumor** across the network

Problem 2

Voting and spreading the result

Problem 3

Reaching consensus on the **average** of the initial values of the agents

History & Motivation

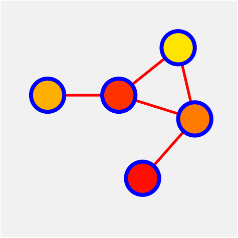
Mathematical formulation

- Pb. 1: Rumor Spreading

Pb. 2: Voting

Pb. 3: Average Consensus
    - the Synchronous Case
    - the Asynchronous Case
    - Communications and averaging

Problem: Each agent has a *rumor* a scalar
the goal is to *spread* the *best one* the maximum

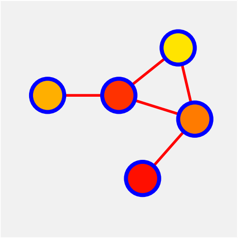


Different setups:

- Does the agent know it has the best rumor?
- Are communications synchronous?
- Which is the communication scheme?

Applications:
Localization, Meteorology, Database management

- Overview of different algorithms with some basic results/theory and pointers



Random Walk

At time k, let i be the active node:

- i sends x_i^k to a neighbor j uniformly chosen in $\mathcal{N}_i$;
- j updates: $x_j^{k+1} = \max\left(x_i^k, x_j^k\right)$;
- j is then the active node for time $k+1$.

Convergence:

Rumor aware (or not)
Synchronous
No Collisions

$\tau \leq H + C$

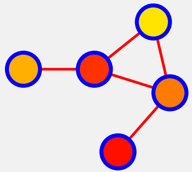
H and C are the maximal hitting time and cover times on the graph

References: Algebraic graph theory and Markov chains

Feige *a tight upper bound on the cover time for random walks on graphs* Random Structures and Algorithms,1995

Aldous & Fill *reversible Markov chains and random walks on graphs*, book draft

Avin & Ercal *one the cover time and mixing time of RGG* Theoretical Computer Science, 2007



Random Pairwise

At time k, let i be the active node:

- i chooses a neighbor j uniformly chosen in $\mathcal{N}_i$
- i and j exchange their values
- i and j updates: $x_i^{k+1} = x_j^{k+1} = \max\left(x_i^k, x_j^k\right)$

Convergence:

a. Rumor unaware, Asynchronous, No Collisions

$$\mathbb{E}[\tau] \leq N d_{\max} \frac{H_{N-1}}{\lambda_2(L)}$$

$$\tau = \mathcal{O}\left(\frac{N \log^2(N) d_{\max}}{\alpha}\right)$$

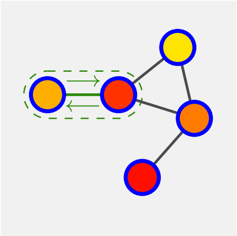
b. Rumor aware, Synchronous, Collisions

$$\tau = \mathcal{O}\left(\frac{\log^{2.5}(N) d_{\max}}{\alpha}\right)$$

$d_{\max}$ is the maximal degree, H_N is the N-th harmonic number,
$\lambda_2(L)$ is the second smallest eig. of the Laplacian, α is the vertex expansion of the graph

References: Algebraic graph theory and Probabilities

I., Ciblat & Jakubowicz *Analysis of max-consensus algorithms in wireless channels* IEEE TSP, 2012
Giakkoupis & Sauerwald *Rumor Spreading and Vertex expansion* SODA, 2012



Random Pairwise

At time k, let i be the active node:

- i chooses a neighbor j uniformly chosen in $\mathcal{N}_i$
- i and j exchange their values
- i and j updates: $x_i^{k+1} = x_j^{k+1} = \max\left(x_i^k, x_j^k\right)$

Convergence:

a. Rumor unaware, Asynchronous, No Collisions

$$\mathbb{E}[\tau] \leq N d_{\max} \frac{H_{N-1}}{\lambda_2(L)}$$

$$\tau = \mathcal{O}\left(\frac{N \log^2(N) d_{\max}}{\alpha}\right)$$

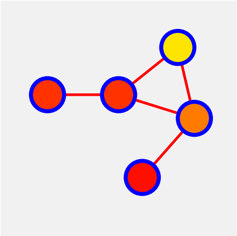
b. Rumor aware, Synchronous, Collisions

$$\tau = \mathcal{O}\left(\frac{\log^{2.5}(N) d_{\max}}{\alpha}\right)$$

$d_{\max}$ is the maximal degree, H_N is the N-th harmonic number,
$\lambda_2(L)$ is the second smallest eig. of the Laplacian, α is the vertex expansion of the graph

References: Algebraic graph theory and Probabilities

I., Ciblat & Jakubowicz *Analysis of max-consensus algorithms in wireless channels* IEEE TSP, 2012
Giakkoupis & Sauerwald *Rumor Spreading and Vertex expansion* SODA, 2012



Random Pairwise

At time k, let i be the active node:

- i chooses a neighbor j uniformly chosen in $\mathcal{N}_i$
- i and j exchange their values
- i and j updates: $x_i^{k+1} = x_j^{k+1} = \max\left(x_i^k, x_j^k\right)$

Convergence:

a. Rumor unaware, Asynchronous, No Collisions

$$\mathbb{E}[\tau] \leq N d_{\max} \frac{H_{N-1}}{\lambda_2(L)}$$

$$\tau = \mathcal{O}\left(\frac{N \log^2(N) d_{\max}}{\alpha}\right)$$

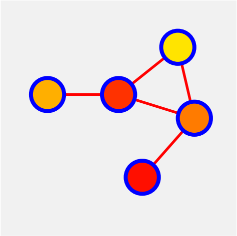
b. Rumor aware, Synchronous, Collisions

$$\tau = \mathcal{O}\left(\frac{\log^{2.5}(N) d_{\max}}{\alpha}\right)$$

$d_{\max}$ is the maximal degree, H_N is the N-th harmonic number,
$\lambda_2(L)$ is the second smallest eig. of the Laplacian, α is the vertex expansion of the graph

References: Algebraic graph theory and Probabilities

I., Ciblat & Jakubowicz *Analysis of max-consensus algorithms in wireless channels* IEEE TSP, 2012
Giakkoupis & Sauerwald *Rumor Spreading and Vertex expansion* SODA, 2012



Random Broadcast

At time k, let i be the active node:

- i broadcasts x_i^k to all its neighbors
- Each neighbor j updates: $x_j^{k+1} = \max\left(x_i^k, x_j^k\right)$

Convergence:

a. Rumor unaware, Asynchronous, No Collisions

$$\mathbb{E}[\tau] \leq N\Delta + N(\Delta - 1)\log\left(\frac{N-1}{\Delta-1}\right)$$

$\tau = \mathcal{O}\left(N \log(N)\right)$ complete graph

b. Rumor aware, Synchronous, Collisions

$$\mathbb{E}[\tau] \leq \Delta \log\left(\frac{N}{\Delta}\right)$$

$\tau = \mathcal{O}\left(\log(N)\right)$ complete graph

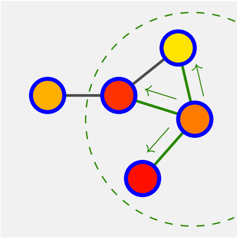
Δ is the diameter of the graph

References: Algebraic graph theory and Probabilities

I., Ciblat & Jakubowicz *Analysis of max-consensus algorithms in wireless channels* IEEE TSP, 2012

Feige et al. *Randomized Broadcast in Networks* Random Structures and Algorithms, 1990

Czumaj & Rytter *Broadcasting algorithms in radio networks* FOCS, 2003



Random Broadcast

At time k, let i be the active node:

- i broadcasts x_i^k to all its neighbors
- Each neighbor j updates: $x_j^{k+1} = \max\left(x_i^k, x_j^k\right)$

Convergence:

a. Rumor unaware / Asynchronous / No Collisions

$\mathbb{E}[\tau] \leq N\Delta + N(\Delta - 1)\log\left(\frac{N-1}{\Delta-1}\right)$

$\tau = \mathcal{O}\left(N\log(N)\right)$ complete graph

b. Rumor aware / Synchronous / Collisions

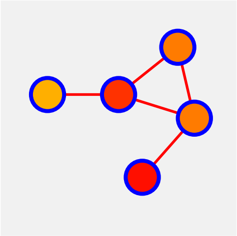
$\mathbb{E}[\tau] \leq \Delta\log\left(\frac{N}{\Delta}\right)$

$\tau = \mathcal{O}\left(\log(N)\right)$ complete graph

Δ is the diameter of the graph

References: Algebraic graph theory and Probabilities

I., Ciblat & Jakubowicz *Analysis of max-consensus algorithms in wireless channels* IEEE TSP, 2012
Feige et al. *Randomized Broadcast in Networks* Random Structures and Algorithms, 1990
Czumaj & Rytter *Broadcasting algorithms in radio networks* FOCS, 2003



Random Broadcast

At time k, let i be the active node:

- i broadcasts x_i^k to all its neighbors
- Each neighbor j updates: $x_j^{k+1} = \max\left(x_i^k, x_j^k\right)$

Convergence:

a. Rumor unaware, Asynchronous, No Collisions

$$\mathbb{E}[\tau] \leq N\Delta + N(\Delta - 1)\log\left(\frac{N-1}{\Delta-1}\right)$$

$\tau = \mathcal{O}\left(N\log(N)\right)$ complete graph

b. Rumor aware, Synchronous, Collisions

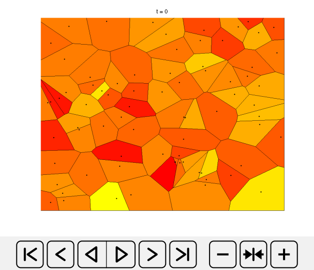
$$\mathbb{E}[\tau] \leq \Delta\log\left(\frac{N}{\Delta}\right)$$

$\tau = \mathcal{O}\left(\log(N)\right)$ complete graph

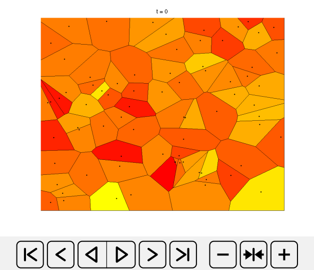
Δ is the diameter of the graph

References: Algebraic graph theory and Probabilities

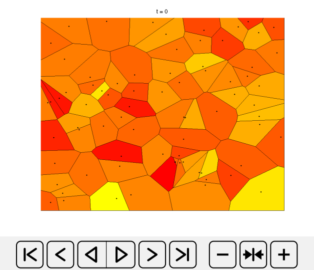
I., Ciblat & Jakubowicz *Analysis of max-consensus algorithms in wireless channels* IEEE TSP, 2012
Feige et al. *Randomized Broadcast in Networks* Random Structures and Algorithms, 1990
Czumaj & Rytter *Broadcasting algorithms in radio networks* FOCS, 2003



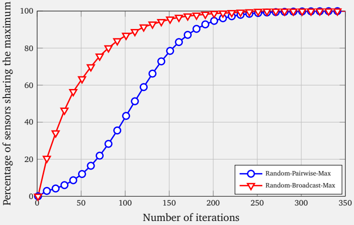
Random Walk



Pairwise



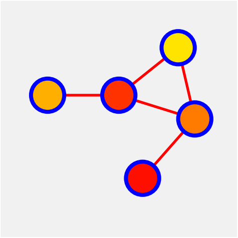
BroadCast



- Factor N in convergence time if the nodes do not known which rumor to spread
- Broadcast communications are very efficient for spreading a rumor
- Tools and Analyses coming from various communities

- History & Motivation
- Mathematical formulation
- Pb. 1: Rumor Spreading
- Pb. 2: Voting
- Pb. 3: Average Consensus
  - the Synchronous Case
  - the Asynchronous Case
  - Communications and averaging

Problem: Each agent has a *vote* binary 0/1
the goal is to *spread* the *majority vote* everyone is aware of the result



Difficult problem:

- impossible by keeping one bit per agent
- 2 bits ok

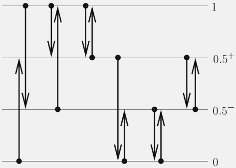
Extensions:
Quorum (e.g. 2/3 majority), multiple choices

- Original algorithm with a beautiful construction

Distributed Voting

At time k, let i be the active node:

- i chooses a neighbor j uniformly chosen in $\mathcal{N}_i$
- i and j exchange their values
- i and j update as given by the figure



- Asynchronous
  Pairwise
  No Collisions

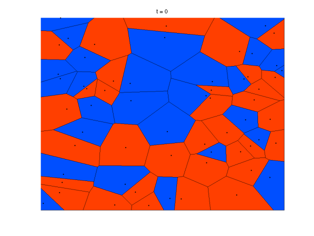
finite time if the graph is connected

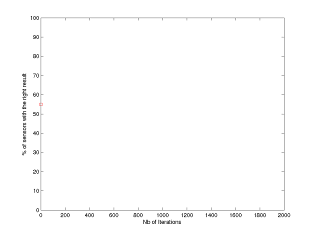
$\tau = \mathcal{O}(N^4 \log(N))$

References: Conservation of sum and Geometry

Benezit, Thiran & Vetterli *The Distributed Multiple Voting Problem* IEEE JSTSP, 2013

Shang et al. *An Upper Bound on the Convergence Time for Distributed Binary Consensus* ArXiv, 2013



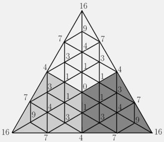


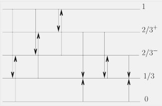
60 agents on a RGG. 33 voted *1 - Red*.

: 0
: 0.5^-
Vote 0 - Blue

: 0.5^+
: 1
Vote 1 - Red

- Original Algorithm with nice formulation
- Extensions to quorum and multiple votes





(c) 2/3 quorum checker

images: Benezit et al.

- History & Motivation
- Mathematical formulation
- Pb. 1: Rumor Spreading
- Pb. 2: Voting
- **Pb. 3: Average Consensus**
  - the Synchronous Case
  - the Asynchronous Case
  - Communications and averaging

History & Motivation

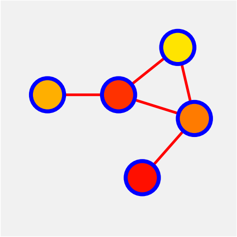
Mathematical formulation

Pb. 1: Rumor Spreading

Pb. 2: Voting

- Pb. 3: Average Consensus
  - the Synchronous Case
  - the Asynchronous Case
  - Communications and averaging

Problem: Each agent has a *value* real (to simplify)
the goal is to *reach a consensus* over the *means value*



Central problem:

- at the heart of *"distributed data/common objective"* pb.
  Detection, LMS, Optimization
- most investigated

Linear updates conditioned by an underlying **communication graph**
for the graph above, it produces the following *mask*

$$x^{k+1} = \underbrace{\begin{bmatrix} \square & \square & 0 & 0 & 0 \\ \square & \square & \square & \square & 0 \\ 0 & \square & \square & \square & 0 \\ 0 & \square & \square & \square & \square \\ 0 & 0 & 0 & \square & \square \end{bmatrix}}_{W^k} x^k$$

- First let us be *synchronous* ($W^k = W$) and investigate desirable properties

Property 1: *Stochasticity*

- non-negative coefficients
- row sum is 1

Algo. interpretation:

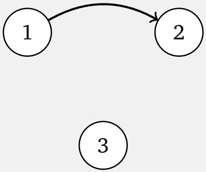
. Each operation is a linear convex combination of the neighbors values

$$\begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0.25 & 0.25 & 0.25 & 0.25 & 0 \\ 0 & 0.33 & 0.33 & 0.33 & 0 \\ 0 & 0.25 & 0.25 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0.5 & 0.5 \end{bmatrix}$$

Consequences:

- spectral radius $\rho(W) = 1$
- $\mathbf{1}$ is an eigenvector for eigenvalue 1
- ∞-norm is 1
  in the ℓ_∞-induced or operator norm

if column sum is 1 (instead of row):
– spectral radius $\rho(W) = 1$
– $\mathbf{1}$ is a left eigenvector for eigenvalue 1
– 1-norm is 1



Property 2: *Primitivity*

- $(W)^n$ has positive entries for some $n > 0$

Graph interpretation:

i. Connected (Strongly)

ii. there is some n such that if we can go from i to j in n steps, we can go in $n+1$ steps

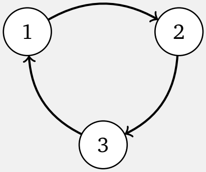
$$W = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

property i. KO

Consequences:

- spectral radius $\rho(W)$ is an eigenvalue
- $\rho(W)$ is the only eigenvalue with maximal modulus

– if the diagonal values are non-zero (or even just one): Connected (Strongly) $\Leftrightarrow$ Primitive
– we do not care about the value of the entries as long as they are positive!
– if there is are *cycles* that contradict the reachability condition, there is as many maximal eigenvalues as the size of the cycle!



Property 2: *Primitivity*

- $(W)^n$ has positive entries for some $n > 0$

Graph interpretation:

i. Connected (Strongly)

ii. there is some n such that if we can go from i to j in n steps, we can go in $n+1$ steps

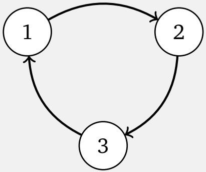
$$W = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

property ii. KO

Consequences:

- spectral radius $\rho(W)$ is an eigenvalue
- $\rho(W)$ is the only eigenvalue with maximal modulus

– if the diagonal values are non-zero (or even just one): Connected (Strongly) $\Leftrightarrow$ Primitive
– we do not care about the value of the entries as long as they are positive!
– if there is are *cycles* that contradict the reachability condition, there is as many maximal eigenvalues as the size of the cycle!



$$W^2 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

property ii. KO

Property 2: *Primitivity*

- $(W)^n$ has positive entries for some $n > 0$

Graph interpretation:

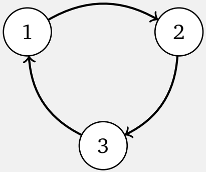
i. Connected (Strongly)

ii. there is some n such that if we can go from i to j in n steps, we can go in $n + 1$ steps

Consequences:

- spectral radius $\rho(W)$ is an eigenvalue
- $\rho(W)$ is the only eigenvalue with maximal modulus

– if the diagonal values are non-zero (or even just one): Connected (Strongly) $\Leftrightarrow$ Primitive
– we do not care about the value of the entries as long as they are positive!
– if there is are *cycles* that contradict the reachability condition, there is as many maximal eigenvalues as the size of the cycle!



Property 2: *Primitivity*

- $(W)^n$ has positive entries for some $n > 0$

Graph interpretation:

i. Connected (Strongly)

ii. there is some n such that if we can go from i to j in n steps, we can go in $n+1$ steps

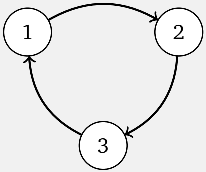
$$W^3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

property ii. KO

Consequences:

- spectral radius $\rho(W)$ is an eigenvalue
- $\rho(W)$ is the only eigenvalue with maximal modulus

– if the diagonal values are non-zero (or even just one): Connected (Strongly) $\Leftrightarrow$ Primitive
– we do not care about the value of the entries as long as they are positive!
– if there is are *cycles* that contradict the reachability condition, there is as many maximal eigenvalues as the size of the cycle!



Property 2: *Primitivity*

- $(W)^n$ has positive entries for some $n > 0$

Graph interpretation:

i. Connected (Strongly)
ii. there is some n such that if we can go from i to j in n steps, we can go in $n+1$ steps

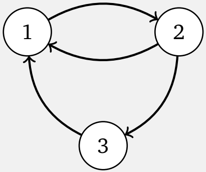
$$W^4 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

property ii. KO

Consequences:

- spectral radius $\rho(W)$ is an eigenvalue
- $\rho(W)$ is the only eigenvalue with maximal modulus

– if the diagonal values are non-zero (or even just one): Connected (Strongly) $\Leftrightarrow$ Primitive
– we do not care about the value of the entries as long as they are positive!
– if there is are *cycles* that contradict the reachability condition, there is as many maximal eigenvalues as the size of the cycle!



$$W = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

primitive

Property 2: *Primitivity*

- $(W)^n$ has positive entries for some $n > 0$

Graph interpretation:

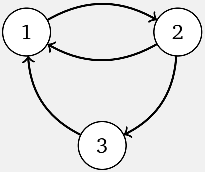
i. Connected (Strongly)

ii. there is some n such that if we can go from i to j in n steps, we can go in $n+1$ steps

Consequences:

- spectral radius $\rho(W)$ is an eigenvalue
- $\rho(W)$ is the only eigenvalue with maximal modulus

– if the diagonal values are non-zero (or even just one): Connected (Strongly) $\Leftrightarrow$ Primitive
– we do not care about the value of the entries as long as they are positive!
– if there is are *cycles* that contradict the reachability condition, there is as many maximal eigenvalues as the size of the cycle!



Property 2: *Primitivity*

- $(W)^n$ has positive entries for some $n > 0$

Graph interpretation:

i. Connected (Strongly)

ii. there is some n such that if we can go from i to j in n steps, we can go in $n+1$ steps

$$W^5 = \begin{bmatrix} 2 & 1 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

primitive

Consequences:

- spectral radius $\rho(W)$ is an eigenvalue
- $\rho(W)$ is the only eigenvalue with maximal modulus

– if the diagonal values are non-zero (or even just one): Connected (Strongly) $\Leftrightarrow$ Primitive
– we do not care about the value of the entries as long as they are positive!
– if there is are *cycles* that contradict the reachability condition, there is as many maximal eigenvalues as the size of the cycle!

Assume that W is ***stochastic* and *primitive***.

1 is the only eigenvalue of greatest modulus. Let λ be the second eigenvalue in modulus, $|\lambda| < 1$.

- let us **iterate** this matrix, looking at its Jordan normal form:

$$x^1 = P \begin{bmatrix} 1 & & & & & \\ & \lambda & 1 & & & \\ & & \ddots & \ddots & & \\ & & & \lambda & 1 & \\ & & & & \lambda & \\ & & & & & \ddots \end{bmatrix} P^{-1} x^0$$

Assume that W is ***stochastic* and *primitive***.
1 is the only eigenvalue of greatest modulus. Let λ be the second eigenvalue in modulus, $|\lambda| < 1$.

- let us **iterate** this matrix, looking at its Jordan normal form:

$$x^k = P \begin{bmatrix} 1 & & & & & \\ & \lambda^k & k\lambda^{k-1} & \dots & \binom{k}{s}\lambda^{k-s} & \\ & & \ddots & \ddots & \vdots & \\ & & & \lambda^k & k\lambda^{k-1} & \\ & & & & \lambda^k & \\ & & & & & \ddots \end{bmatrix} P^{-1}x^0$$

Convergence:

- Exponential in $\tilde{\mathcal{O}}(\lambda^k)$
- to a rank-1 matrix uv^{T}
  u and v are the right and left eigenvectors associated with 1

True for row-sto. ($u = \mathbf{1}$) and col.-sto. ($v = \mathbf{1}$)

Assume that W is ***row-stochastic* and *primitive***.

1 is the only eigenvalue of greatest modulus. Let λ be the second eigenvalue in modulus, $|\lambda| < 1$.

- let us **iterate** this matrix, looking at its Jordan normal form:

$$x^k \to \mathbf{1}v^{\mathrm{T}}x^0$$

as $\mathbf{1}$ is the eigenvector associated with eigenvalue 1
v is the corresponding left eigenvector; $v^{\mathrm{T}}\mathbf{1} = 1$

Assume that W is ***column-stochastic* and *primitive***.

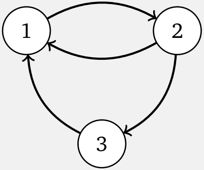
1 is the only eigenvalue of greatest modulus. Let λ be the second eigenvalue in modulus, $|\lambda| < 1$.

- let us **iterate** this matrix, looking at its Jordan normal form:

$$x^k \to v\mathbf{1}^\mathrm{T}x^0$$

as $\mathbf{1}$ is the *left* eigenvector associated with eigenvalue 1
v is the corresponding *(right)* eigenvector; $v^\mathrm{T}\mathbf{1} = 1$

In PageRank, that is exactly what is done. v is the Perron Vector.
The use of $R = (1-\alpha)W + \alpha J$ enables to retrieve primitivity!



$$W = \begin{bmatrix} 0 & 1 & 0 \\ 0.5 & 0 & 1 \\ 0.5 & 0 & 0 \end{bmatrix}$$

eig.: $1, 0.5 \pm 0.5i$

modulus: $1, 0.707$

$$W^{20} = \begin{bmatrix} 0.40 & 0.40 & 0.40 \\ 0.40 & 0.40 & 0.40 \\ 0.20 & 0.20 & 0.20 \end{bmatrix}$$

Assume that W is ***doubly-stochastic* and *primitive***.
1 is the only eigenvalue of greatest modulus. Let λ be the second eigenvalue in modulus, $|\lambda| < 1$.

- let us **iterate** this matrix, looking at its Jordan normal form:

$$x^k \to \mathbf{1}\mathbf{1}^{\mathrm{T}} \frac{1}{N} x^0 = \mathbf{1} x_{\text{ave}}$$

as $\mathbf{1}$ is an eigenvector associated with eigenvalue 1 at both sides
$1/N$ comes from the joint normalization.

Synchronous Average Consensus:

- Doubly-Stochastic Primitive matrix
- Exponential rate depending on the second eigenvalue $\approx$ the algebraic connectivity

ex. 1:
full graph
one step consensus

$$J = \frac{1}{N} \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}$$

ex. 2:
connected graph
Metropolis-Hastings

$$W_{i,j} = \begin{cases} \frac{1}{1+\max(d_i, d_j)} & \text{if } j \in \mathcal{N}_i \\ 1 - \sum_{j \in \mathcal{N}_i} W_{i,j} & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

- Let us get to *randomized* average gossip! (after a small parenthesis)

Differential equations can be used to prove convergence and rates at large
gradient, accelerated gradient, FISTA

Define the $|\mathcal{E}| \times N$ *incidence matrix* of a graph:
□ $M_{ev} = 1$ if $e = (v, j)$; -1 if $e = (i, v)$; 0 elsewhere.
□ M is a square root of L: $L = M^{\mathrm{T}} M$!

Consider reaching consensus by minimizing $f(x) = \frac{1}{2} \|Mx\|^2$ and observe the ODE:

$$\dot{x}(t) = -\nabla f(x(t)) = -M^{\mathrm{T}} M x(t) = -Lx(t)$$

An Euler discretisation gives:

$$\frac{x^{k+1} - x^k}{\gamma} = -Lx^k \Leftrightarrow x^{k+1} = x^k - \gamma L x^k = \underbrace{(I - \gamma L)}_{W} x^k$$

It is easy to see that with step $\gamma < 1/d_{\max}$, W is non-negative, doubly stochastic, primitive: it is an averaging algorithm.

References Pseudo-orbit of ODEs, Inertia, Krause model dynamical graph, multiple clusters

Attouch, Peypouquet, & Redont *A dynamical approach to an inertial forward-backward algorithm for convex minimization*, SIOPT, 2014.
Blondel, Hendrickx, & Tsitsiklis *On Krause's multi-agent consensus model with state-dependent connectivity*, IEEE TAC, 2009.

History & Motivation

Mathematical formulation

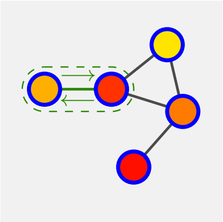
Pb. 1: Rumor Spreading

Pb. 2: Voting

- Pb. 3: Average Consensus
  - the Synchronous Case
  - the Asynchronous Case
  - Communications and averaging

Problem: Each agent has a *value* real (to simplify)
the goal is to *reach a consensus* over the *means value*

Linear updates conditioned by an underlying **communication graph**
for the graph above, it produces the following *mask*



$$x^{k+1} = \underbrace{\begin{bmatrix} \square & \square & 0 & 0 & 0 \\ \square & \square & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{W^k} x^k$$

Changes compared to Synchronous case:

- At each iteration, a mixing matrix is randomly chosen in a set
- The objective is that only some nodes/links are involved at each iteration

Standard Model:

- Set $\mathcal{W} = \{W_1, ..., W_M\}$ of M ***doubly-stochastic*** *update matrices*
- **i.i.d. process** $\{\xi^k\}_{k>0}$ valued in $\{1, ..., M\}$
- $\mathbb{E}[W]$ is ***primitive***

$$x^{k+1} = W_{\xi^{k+1}} x^k = W_{\xi^{k+1}} W_{\xi^k} \dots W_{\xi^1} x^0$$

Convergence:

- $x^k \to \mathbf{1} x_{ave}$ almost surely
- The mean squared error vanishes exponentially
- Exact rate $\rho\left(\mathbb{E}[W \otimes W].\left((I - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}}) \otimes (I - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}})\right)\right)$
  outside a set of Lebesgue measure 0

References: Matrix Analysis and Probabilities

Boyd et al. *Randomized gossip algorithms* IEEE/ACM Transactions on Networking,2006
I. & Ciblat & Hachem *Analysis of Sum-Weight-like algorithms for averaging in Wireless Sensor Networks* IEEE TSP,2013

Classic way quick but not tight

$$
\begin{aligned}
\|(I-J)x^{k+1}\|_2^2 &= \|(I-J)W_{\xi^{k+1}}(I-J)x^k\|_2^2 \text{(row-sto.)} \\
&= ((I-J)x^k)^{\mathrm{T}}((I-J)W_{\xi^{k+1}})^{\mathrm{T}}(I-J)W_{\xi^{k+1}}(I-J)x^k
\end{aligned}
$$

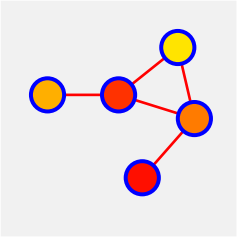
$$
\begin{aligned}
\mathbb{E}[\|(I-J)x^{k+1}\|_2^2|\mathcal{F}_k] &= ((I-J)x^k)^{\mathrm{T}}\mathbb{E}[((I-J)W_{\xi^{k+1}})^{\mathrm{T}}(I-J)W_{\xi^{k+1}}|\mathcal{F}_k](I-J)x^k \\
&\leq \underbrace{\rho\left(\mathbb{E}[((I-J)W_{\xi^{k+1}})^{\mathrm{T}}(I-J)W_{\xi^{k+1}}|\mathcal{F}_k]\right)}_{:=\sigma<1}\|(I-J)x^k\|_2^2
\end{aligned}
$$

Alternative way tight

$$
\begin{aligned}
\mathbb{E}[[(I-J)x^{k+1}]\otimes[(I-J)x^{k+1}]|\mathcal{F}_k] &= \mathbb{E}[[(I-J)W_{\xi^{k+1}}]\otimes[(I-J)W_{\xi^{k+1}}]|\mathcal{F}_k] \\
&\quad \times [(I-J)x^k]\otimes[(I-J)x^k] \\
&\propto \underbrace{\rho\left(\mathbb{E}[[(I-J)W_{\xi^{k+1}}]\otimes[(I-J)W_{\xi^{k+1}}]|\mathcal{F}_k]\right)^k}_{:=\varsigma<1}
\end{aligned}
$$

- MSE converges exponentially to 0
- Markov + Borel-Cantelli = Almost-sure convergence

$J = 1/N\mathbf{1}\mathbf{1}^{\mathrm{T}}$

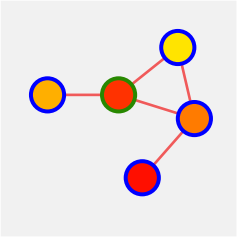


Random Gossip

At time k, let i be the active node:

Features:

– Two nodes and one link used at each iteration
– Introduced by Boyd et al. in 2004
– Converges as soon as the (undirected) graph is connected



Random Gossip

At time k, let i be the active node:

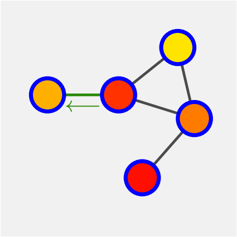
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Features:

– Two nodes and one link used at each iteration

– Introduced by Boyd et al. in 2004

– Converges as soon as the (undirected) graph is connected



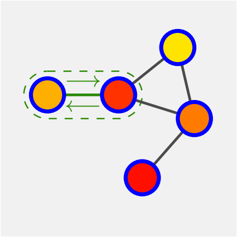
Random Gossip

At time k, let i be the active node:

- i chooses a neighbor j uniformly in $\mathcal{N}_i$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Features:
– Two nodes and one link used at each iteration
– Introduced by Boyd et al. in 2004
– Converges as soon as the (undirected) graph is connected



Random Gossip

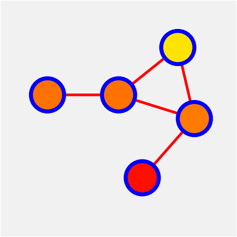
At time k, let i be the active node:

- i chooses a neighbor j uniformly in $\mathcal{N}_i$
- they exchange their values

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Features:

– Two nodes and one link used at each iteration
– Introduced by Boyd et al. in 2004
– Converges as soon as the (undirected) graph is connected



Random Gossip

At time k, let i be the active node:

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

$$\begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Features:

– Two nodes and one link used at each iteration
– Introduced by Boyd et al. in 2004
– Converges as soon as the (undirected) graph is connected

Under the standard assumptions, convergence to average consensus is easily obtained

- Primitivity in average is *natural*
- Double-stochasticity at each iteration is *mandatory*

Other formulation for asynchronous average gossip

- The product of any $B < \infty$ consecutive matrices is positive
  sort of *deterministic asynchrony*
  kind of asynchrony in [Tsitsiklis'84]

Double-stochasticity constrains the coefficients of the mixing matrix and thus the communicated values

- Let us investigate different communications schemes

- History & Motivation
- Mathematical formulation
- Pb. 1: Rumor Spreading
- Pb. 2: Voting
- Pb. 3: Average Consensus
  - the Synchronous Case
  - the Asynchronous Case
  - Communications and averaging

We saw earlier the good properties of ***broadcast* communications**

How to design a mixing matrix when **agent** 5 ***broadcasts*** to its neighbors 2, 3 and 4

- no feedback

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & w_{2,2} & 0 & 0 & w_{2,5} \\ 0 & 0 & w_{3,3} & 0 & w_{3,5} \\ 0 & 0 & 0 & w_{4,4} & w_{4,5} \\ 0 & 0 & 0 & 0 & w_{5,5} \end{bmatrix}$$

We saw earlier the good properties of ***broadcast* communications**

How to design a mixing matrix when **agent** 5 ***broadcasts*** to its neighbors 2, 3 and 4

- no feedback
- Row-stochasticity is possible...

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & w_2 & 0 & 0 & 1-w_2 \\ 0 & 0 & w_3 & 0 & 1-w_3 \\ 0 & 0 & 0 & w_4 & 1-w_4 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

We saw earlier the good properties of ***broadcast* communications**

How to design a mixing matrix when **agent** 5 ***broadcasts*** to its neighbors 2, 3 and 4

- no feedback
- Row-stochasticity is possible...
- ... but not double-stochasticity

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & w_2 & 0 & 0 & 1-w_2 \\ 0 & 0 & w_3 & 0 & 1-w_3 \\ 0 & 0 & 0 & w_4 & 1-w_4 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

We saw earlier the good properties of ***broadcast* communications**

How to design a mixing matrix when **agent** 5 ***broadcasts*** to its neighbors 2, 3 and 4

- no feedback
- Row-stochasticity is possible...
- ... but not double-stochasticity
- Column-stochasticity is possible

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & w_{2,2} & 0 & 0 & w_{2,5} \\ 0 & 0 & w_{3,3} & 0 & w_{3,5} \\ 0 & 0 & 0 & w_{4,4} & w_{4,5} \\ 0 & 0 & 0 & 0 & w_{5,5} \end{bmatrix}$$

We saw earlier the good properties of ***broadcast* communications**

How to design a mixing matrix when **agent** 5 ***broadcasts*** to its neighbors 2, 3 and 4

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & w_{2,5} \\ 0 & 0 & 1 & 0 & w_{3,5} \\ 0 & 0 & 0 & 1 & w_{4,5} \\ 0 & 0 & 0 & 0 & w_{5,5} \end{bmatrix}$$

- no feedback
- Row-stochasticity is possible...
- ... but not double-stochasticity
- Column-stochasticity is possible

  $$w_{2,5} + w_{3,5} + w_{4,5} + w_{5,5} = 1$$

- We need to choose between **row** and **column** stochasticity

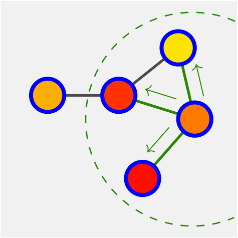
Model:

- Set $\mathcal{W} = \{W_1, \ldots, W_M\}$ of M ***non-negative*** *update matrices*
- **i.i.d. process** $\{\xi^k\}_{k>0}$ valued in $\{1, \ldots, M\}$
- $\mathbb{E}[W]$ is ***primitive***

$$x^{k+1} = W_{\xi^{k+1}} x^k = W_{\xi^{k+1}} W_{\xi^k} \ldots W_{\xi^1} x^0$$

row sto.	**column sto.**
$W_{\xi^{k+1}} W_{\xi^k} \ldots W_{\xi^1} \to \mathbf{1} v^{\mathrm{T}}(\omega)$	$W_{\xi^{k+1}} W_{\xi^k} \ldots W_{\xi^1} \sim v^k(\omega) \mathbf{1}^{\mathrm{T}} \frac{1}{N}$
$x^k \to \mathbf{1}(v^{\mathrm{T}}(\omega) x^0)$	$x^k \sim v^k(\omega)(\mathbf{1}^{\mathrm{T}} \frac{1}{N} x^0) = v^k x_{ave}$
Consensus Wrong Value	No Convergence Good mean value ($\sum_i v_i^k(\omega) = 1$)

$$x^k \to \mathbf{1}(\nu^{\mathrm{T}}(\omega)x^0)$$



Broadcast Gossip

At each clock tick k, let i be the activating node:

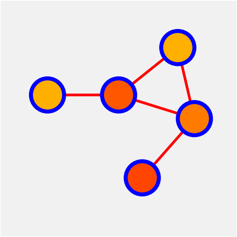
- i broadcasts x_i^k to all its neighbors
- All the neighbors update: $x_j^{k+1} = \frac{x_i^k + x_j^k}{2}$ for all $j \in \mathcal{N}_i$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Features:

- $\mathbb{E}_\omega[\nu(\omega)] = \frac{1}{N}\mathbf{1}$
- Analyzed by Aysal et al. in 2009

$$x^k \to \mathbf{1}(v^{\mathrm{T}}(\omega)x^0)$$



BROADCAST GOSSIP

At each clock tick k, let i be the activating node:

- i broadcasts x_i^k to all its neighbors
- All the neighbors update: $x_j^{k+1} = \frac{x_i^k + x_j^k}{2}$ for all $j \in \mathcal{N}_i$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0.5 \\ 0 & 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Features:

- $\mathbb{E}_\omega[v(\omega)] = \frac{1}{N}\mathbf{1}$
- Analyzed by Aysal et al. in 2009

$$s^k = W_{\xi^k} W_{\xi^{k-1}} ... W_{\xi^1} x^0 \sim v^k (\mathbf{1}^{\mathrm{T}} \frac{1}{N} x^0) = v^k x_{ave}$$

$$s^k = W_{\xi^k} W_{\xi^{k-1}} ... W_{\xi^1} x^0 \sim v^k (\mathbf{1}^{\mathrm{T}} \frac{1}{N} x^0) = v^k x_{ave}$$

Let us define: $w^k = W_{\xi^k} W_{\xi^{k-1}} ... W_{\xi^1} \mathbf{1} \sim v^k (\mathbf{1}^{\mathrm{T}} \frac{1}{N} \mathbf{1}) = v^k$

$$s^k = W_{\xi^k} W_{\xi^{k-1}} \dots W_{\xi^1} x^0 \sim v^k (\mathbf{1}^{\mathrm{T}} \frac{1}{N} x^0) = v^k x_{ave}$$

$$\text{Let us define: } w^k = W_{\xi^k} W_{\xi^{k-1}} \dots W_{\xi^1} \mathbf{1} \sim v^k (\mathbf{1}^{\mathrm{T}} \frac{1}{N} \mathbf{1}) = v^k$$

Sum-Weight framework:

- Two variables updated in the same way but initialized differently
  - A *sum* variable initialized with x^0
  - A *weight* variable initialized with $\mathbf{1}$
- The estimate is

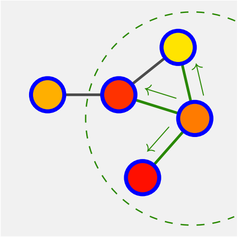
$$x^k \triangleq \left[\frac{s_1^k}{w_1^k}, \dots, \frac{s_N^k}{w_N^k} \right] = \frac{s^k}{w^k} \sim \frac{v^k x_{ave}}{v^k} = x_{ave} \mathbf{1}$$

Convergence:

- Almost sure convergence for sequences of column-sto. matrices with $\mathbb{E}[K]$ primitive
  Difficulty: (w^k) is not bounded away from 0...
- Tight linear rate $\rho \left(\mathbb{E}[W \otimes W]. \left((I - \frac{1}{N} \mathbf{1}\mathbf{1}^{\mathrm{T}}) \otimes (I - \frac{1}{N} \mathbf{1}\mathbf{1}^{\mathrm{T}}) \right) \right)$
  Same as for the doubly sto. case

References: Matrix Analysis, Probabilities on graphs

Kempe et al. *Gossip-based computation of aggregate information* FOCS, 2003.
Benezit et al. *Weighted gossip: Distributed averaging using non-doubly stochastic matrices* ISIT, 2010.
I., Ciblat & Hachem *Analysis of Sum-Weight-like algorithms for averaging in WSNs*, IEEE TSP, 2013.



BWGOSSIP

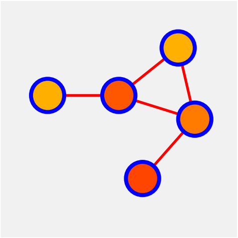
At each clock tick k, let i be the activating node:

- i broadcasts $\left(\frac{s_i^k}{1+d_i}, \frac{w_i^k}{1+d_i}\right)$ to its neighbors
- All the neighbors update: $s_j^{k+1} = s_j^k + \frac{s_i^k}{1+d_i}$ and $w_j^{k+1} = w_j^k + \frac{w_i^k}{1+d_i}$ for all $j \in \mathcal{N}_i$
- i updates: $s_i^{k+1} = \frac{s_i^k}{1+d_i}$ and $w_i^{k+1} = \frac{w_i^k}{1+d_i}$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Features

- Converges linearly (in L2) to the average
- Only Broadcast communications (no feedback)
- Introduced in 2012



BWGossip

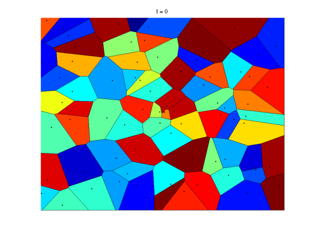
At each clock tick k, let i be the activating node:

- i broadcasts $\left(\frac{s_i^k}{1+d_i}, \frac{w_i^k}{1+d_i}\right)$ to its neighbors
- All the neighbors update: $s_j^{k+1} = s_j^k + \frac{s_i^k}{1+d_i}$ and $w_j^{k+1} = w_j^k + \frac{w_i^k}{1+d_i}$ for all $j \in \mathcal{N}_i$
- i updates: $s_i^{k+1} = \frac{s_i^k}{1+d_i}$ and $w_i^{k+1} = \frac{w_i^k}{1+d_i}$

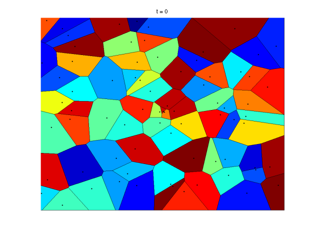
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.25 \\ 0 & 0 & 1 & 0 & 0.25 \\ 0 & 0 & 0 & 1 & 0.25 \\ 0 & 0 & 0 & 0 & 0.25 \end{bmatrix}$$

Features

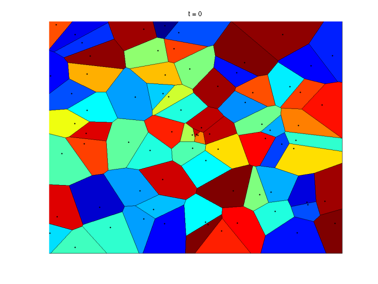
- Converges linearly (in L2) to the average
- Only Broadcast communications (no feedback)
- Introduced in 2012



Random Gossip

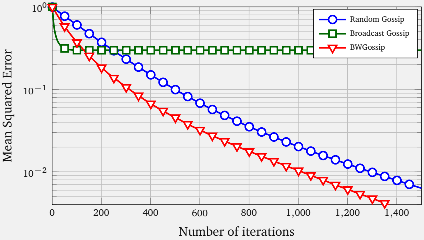


BroadCast Gossip



Sum-Weight BroadCast

: Min
: Max
: Mean



- Randomized gossip is easy with symmetric communications
- For one-way communications, convergence can be obtain by using side-information

Summary:

- Different algorithms/proofs for different objective
- Algebraic graph theory & Markov chains at the center
- The communications scheme plays a great role

END OF PART I