Gossip algorithms for solving Laplacian systems

Anastasios Zouzias
University of Toronto

joint work with Nikolaos Freris (EPFL)

Based on:
1. Fast Distributed Smoothing for Clock Synchronization (CDC ‘12)
2. Randomized Extended Kaczmarz for Solving Least-Squares (Arxiv, May ‘12)
I. Problems: Solving Laplacian & edge-vertex systems
II. Motivation: Clock Synchronization over WSNs
III. Randomized Gossip Model & Averaging Problem
IV. Gossip Solvers via Randomized (Extended) Kaczmarz
I. Problems: Solving Laplacian & edge-vertex systems

II. Motivation: Clock Synchronization over WSNs

III. Randomized Gossip Model & Averaging Problem

IV. Gossip Solvers via Randomized (Extended) Kaczmarz
Distributed solver: Laplacian system

Model of computation
- Each node is aware of its neighbors; exchanges packets with them only
- Static network; no communication errors; ignore numerical issues
- Synchronous, asynchronous & Gossip

Problem 1
Input: Each node $i$ gets $b_i$
Goal: Each node $i$ computes $i^{th}$ coordinate of $x_{LS} := L^\dagger b$
Distributed solver: edge-vertex system

Edge-vertex System

\[
\begin{align*}
G &= (V,E): n \text{ nodes, } m \text{ edges} \\
\text{Problem II} & \quad \text{Input: Each edge } (i,j) \text{ gets } y(i,j) \\
\text{Goal: Each node } i \text{ computes } i^{th} \text{ coordinate of } x_{LS} = B^\dagger y
\end{align*}
\]

edge-vertex incident matrix of G:

\[
B_{ek} := \begin{cases} 
-1, & \text{if } k = i; \\
1, & \text{if } k = j; \\
0, & \text{otherwise.}
\end{cases}
\]

Normal equation of \(Bx = y\) is Laplacian system (\(L = B^T B\))
Outline

I. Problems: Solving Laplacian & edge-vertex Systems

II. Motivation: Clock Synchronization over WSNs

III. Randomized Gossip Model & Averaging Problem

IV. Gossip Solvers via Randomized (Extended) Kaczmarz
Case Study: Clock Synchronization over WSNs

Assumptions

- Each node has clock; same speed
  \( o_v(t) = t + o_v, \quad o_v \in \mathbb{R} \)
- Node \( v \) does not know \( o_v \)
Nodes can approx. relative offsets \( o_{uv} = o_v - o_u \) for every \( u \in \text{Neigh}(v) \)

Clock Synchronization Problem

Input: Estimates \( \hat{o}_{uv} = o_{uv} + \mathcal{N}(0, 1) \) for all \( (u, v) \in E \)

Goal: Compute offsets \( (\tilde{o}_u)_{u \in V} \) that min. \( \max_{u,v} \mathbb{E}|\tilde{o}_{uv} - o_{uv}|^2 \) over all pairs of nodes
**Tree-based Approach**

**Idea:** Build a spanning tree

Path $u \sim v : \text{diam}(G)$

Every edge: normal error

$$\hat{o}_{uv} = \sum_{(u',v') \in P} \hat{o}_{u'v'}$$

Sync error between $u$ & $v$
grows like $\approx O(\sqrt{\text{diam}(G)})$

In general, no hope for better accuracy

...but wireless networks are "well-connected"
Modeling Wireless Networks

...as Random Geometric Graphs

- $n$ nodes uniform over square

- Connectivity $[GK00]$: $r = \mathcal{O}\left(\sqrt{\log n/n}\right)$

- Diameter: $\mathcal{O}\left(\sqrt{n/\log n}\right)$

- Tree-based approach: error $\tilde{O}(n^{1/4})$

Q: Can we do better on Random Geometric Graphs?

Yes! Spatial Smoothing $[KEES03,GK06]$
**Observation:** Every loop in $G$: sum of relative offsets equals zero

**Idea:** Incorporate the loop constraints

**How?**
Encode constraints in linear system:

$$Bx = 0$$

Relative offset of $(i, j)$
Properties of Least-Squares

Gaussian error: compute LS solution of $Bx = \hat{o}$

Thm[KEES03] Replace each edge by unit resistor. Then error variance between any pair of nodes $u$ and $v$ is:

$$E|\tilde{\delta}_{uv} - o_{uv}|^2 \sim R_{\text{eff}}(u, v)$$

Effective resistances of RGG bounded by $O(1)$ [GK06]

Tree-based vs Smoothing
$O(n^{1/4})$ vs $O(1)$

Q: How to compute the LS solution?
The Model Matters...

Use coordinate descent: \[
\frac{\partial}{\partial x_u} \|Bx - o\|^2 = 0
\]

**Synchronous Jacobi**

\[\hat{o}_v = 0, \quad \forall v \in V\]

For \(k = 1, 2, \ldots\)

\[\tilde{o}_v^{(k+1)} \leftarrow \frac{1}{d_v} \sum_{u \in \text{Neigh}(v)} \left( \tilde{o}_u^{(k)} + \hat{o}_{uv} \right)\]

Thm[GK06]: After \(k \geq \frac{4m^2}{\beta^2} \ln(\|x^*\|/\varepsilon)\) rounds, it holds that

\[\|x^{(k)} - x^*\|_2 \leq \varepsilon\]

where \(\beta\) is the min-cut value

**Asynchronous Jacobi**

Each node \(v \in V\) regularly:

- estimates relative offsets with nghbrs
- broadcasts its current offset \(\hat{o}_v\)
- updates its estimate:

\[\tilde{o}_v \leftarrow \frac{1}{d_v} \sum_{u \in \text{Neigh}(v)} \left( \tilde{o}_u + \hat{o}_{uv} \right)\]

*It converges*[BT89]
The Model Matters...

**Randomized Gossip Model**
(a.k.a. asynchronous time model)

[BT89, BGPS06]

Each node $u$ (randomly) activates itself w.p. $p_u$ & performs local computation.

**Synchronous Model**

$\hat{\delta}_v = 0, \quad \forall v \in V$

For $k = 1, 2, \ldots$

$$\hat{\delta}_v^{(k+1)} \leftarrow \frac{1}{d_v} \sum_{u \in \text{Neigh}(v)} \left( \hat{\delta}_u^{(k)} + \hat{\delta}_{uv} \right)$$

Thm[GK06]: After $k \geq \frac{4m^2}{\beta^2} \ln(\|x^*\|/\epsilon)$ rounds, it holds that

$$\|x^{(k)} - x^*\|_2 \leq \epsilon$$

where $\beta$ is the min-cut value

**Asynchronous Model**

Each node $v \in V$ regularly:

- estimates relative offsets with nghbrs
- broadcasts its current offset $\hat{\delta}_v$
- updates its estimate:

$$\hat{\delta}_v \leftarrow \frac{1}{d_v} \sum_{u \in \text{Neigh}(v)} (\hat{\delta}_u + \hat{\delta}_{uv})$$

It converges[BT89]
Outline

I. Problems: Solving Laplacian & edge-vertex systems

II. Motivation: Clock Synchronization over WSNs

III. Randomized Gossip Model & Averaging Problem

IV. Gossip Solvers via Randomized (Extended) Kaczmarz
Distributed Averaging:

**Input**: Every node \( u \) gets \( w_u \)

**Goal**: Every node want access to global average

**Gossip averaging algorithm**

1. Every node \( u \) activates uniformly at random
2. Picks random neighbor \( v \) and averages their current values \( w_u, w_v \)

[BGPS06] proved that \( \mathcal{O}(\frac{n}{\lambda_2(G)} \log(n/\varepsilon)) \) rounds are sufficient whp

Special cases, complete graph [KSSV00,KDG03,KDN+06]

How many rounds required to approx. within \( \varepsilon \)?

Basic primitive for other functions

Averaging can solve Problems I and II

[BDFSV10,XBL05,XBL06]
Gossip Model

Assumptions

• Each node $u$ has independent Poisson time process: rate $\gamma_u$
• Each node activates when its arrival occurs
• Equivalently*: single global Poisson process: rate $\sum_{u \in V} \gamma_u$
• Arrivals correspond to rounds

Claim: Non-uniform sampling of nodes is feasible with zero communication under gossip model (given $\gamma_u$‘s)

Goal: Design and analyze gossip algorithms for Problem I and II

*minimum of ind. Poisson is equivalent to single Poisson with sum of their rates
Outline

I. Problems: Solving Laplacian & edge-vertex systems
II. Motivation: Clock Synchronization over WSNs
III. Randomized Gossip Model & Averaging Problem
IV. Gossip Solvers via Randomized (Extended) Kaczmarz
Kaczmarz Method

Initialize: \( x^{(0)} = 0 \)

Repeat:
Set \( i_k = k \mod m + 1 \)

\[
\begin{align*}
\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \frac{y_{i_k} - \langle A(i_k), \mathbf{x}^{(k)} \rangle}{\|A(i_k)\|^2} A(i_k) \\
k &= k + 1
\end{align*}
\]

(Assumption: \( \mathbf{A} \mathbf{x} = \mathbf{y} \) has solution)

It convergences [K37]

Huge literature; many extensions; rediscovered many times
Randomized Kaczmarz Method

Initialize: $x^{(0)} = 0$

Repeat:
- Pick $i_k \in [m]$ w.p. $p_i \propto \|A^{(i)}\|^2$
- $x^{(k+1)} = x^{(k)} + \frac{y_{i_k} - \langle A^{(i_k)}, x^{(k)} \rangle}{\|A^{(i_k)}\|^2} A^{(i_k)}$
- $k = k + 1$

(Assumption: $Ax=y$ has solution)

Exponential convergence

$\mathbb{E} \left\| x^{(k)} - x^* \right\|^2 \leq \left( 1 - \frac{1}{\kappa_F^2(A)} \right)^k \|x^*\|^2$

where $\kappa_F^2(A) := \frac{\|A\|_F^2}{\sigma_{\text{min}}^2(A)}$
Let’s apply RK on Problems I and II
RK Laplacian Solver

Laplacian System

\[ \begin{bmatrix} L \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} b \end{bmatrix} \]

Randomized Kaczmarz (RK) [SV06]

Initialize: \( x^{(0)} = 0 \)

Repeat:
- Pick \( i_k \in [n] \) w.p. \( p_i \propto \| L^{(i)} \|^2 \)
- \( x^{(k+1)} = x^{(k)} + \frac{b_{i_k} - \langle L^{(i_k)}, x^{(k)} \rangle}{\| L^{(i_k)} \|^2} L^{(i_k)} \)
- \( k = k + 1 \)

RK analysis & diag. preconditioning: \( \tilde{O}(n/\lambda^2(G)) \) rounds whp

Gossip Laplacian Solver

Each node \( u: x_u = 0 \)

Repeat:
- Node \( u \) activates w.p. \( d_u^2 + d_1 \)
- broadcasts \( \theta = x_u - \frac{1}{1 + d_u} \sum_{\ell \in N_u} x_\ell \)
- sets \( x_u \leftarrow x_u + \frac{\theta - b_u/d_u}{1 + d_u} \)
- \( x_v \leftarrow x_v + \frac{\theta - b_u/d_u}{1 + d_u} \)

(Assumption: \( Lx = b \) has solution)
RK Edge-vertex Solver

Consistency assumption (limitation of RK)

How to handle the general case?

Randomized Kaczmarz (RK)

Initialize: $x^{(0)} = 0$

Repeat:

Pick $e = (i, j) \in E$ uniformly

$e = (i, j) \in E$

$B = \begin{bmatrix} 1 & -1 \\ 
\end{bmatrix}$

$m \times n$

$\begin{bmatrix} x \\ 
\end{bmatrix}$

$\begin{bmatrix} B \\ 
\end{bmatrix}$

$y$

$x^{(k+1)} = x^{(k)} + \frac{y_e - \langle B(e), x^{(k)} \rangle}{2} B(e)$

$k = k + 1$

RK analysis & diag. preconditioning:

$\tilde{O}(n/\lambda_2(G))$ rounds whp

Node $u$ activates w.p. $d_u$ & selects random neighbor $v$

- sends $x_u$ & receives $x_v$
- Performs: $x_u \leftarrow (x_u + x_v + y_{(u,v)})/2$

Similarly for node $v$

(Assumption: $Bx = y$ has solution)
Randomized Kaczmarz (RK) [SV06]

**Initialize:** $\mathbf{x}^{(0)} = 0$

**Repeat:**
1. Pick $i_k \in [m]$ w.p. $p_i \propto \|A^{(i)}\|^2$
2. $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \frac{y_{i_k} + w_{i_k} - \langle A^{(i_k)}, \mathbf{x}^{(k)} \rangle}{\|A^{(i_k)}\|^2}A^{(i_k)}$
3. $k = k + 1$

(Assumption: $A\mathbf{x} = \mathbf{y}$ has solution)

**RK is robust to noise** [Needell09]

$\mathbb{E} \left\| \mathbf{x}^{(k)} - \mathbf{x}_{LS} \right\|^2 \leq \left(1 - \frac{1}{\kappa_F^2(A)}\right)^k \|\mathbf{x}_{LS}\|^2 + \frac{\|\mathbf{w}\|^2}{\sigma_{\min}^2(A)}$

Converges to ball centered at LS solution
Robustness of RK: convergence into fixed ball

Idea:
‘Modify’ original system \( Ax = y \) s.t.
1) \( x_{LS} \) is preserved \( \checkmark \)
2) Modified system has LS error \( \approx 0 \) \( \checkmark \)

How?
\[
Ax = y \quad \rightarrow \quad Ax = y_{\mathcal{R}(A)}
\]

Problem
Given subspace as \( \text{colspan}(A) \), vector \( y \) and \( \varepsilon > 0 \)
Goal: Find \( z \) s.t. \( \| z - y_{\mathcal{R}(A)} \| \leq \varepsilon \)
Randomized Orthogonal Projection

**Problem**
Given subspace as $\text{colspan}(A)$, vector $y$ and $\varepsilon > 0$

**Goal:** Find $z$ s.t. $\|z - y_{R(A)}\| \leq \varepsilon$

---

**randOP**

Initialize: $z^{(0)} = y$

Repeat:
- Pick $j_k \in [n]$ w.p. $p_j \propto \|A(j)\|^2$
- $z^{(k+1)} = (I - \frac{A(j_k)A^T(j_k)}{\|A(j_k)\|^2})z^{(k)}$
- $k = k + 1$

---

Exponential convergence $[Z. Freris 12]$:
$$
E \left\| z^{(k)} - y_{R(A)} \right\| \leq \left( 1 - \frac{1}{\kappa^2_F(A)} \right)^k \|y\|^2
$$

Orthogonality gives $y_{R(A)}$
Randomized Extended Kaczmarz for LS

$RK + \text{randOP} =$
Randomized Extended Kaczmarz

**Randomized Extended Kaczmarz**

**Initialize:** $x^{(0)} = 0, z^{(0)} = y$

**Repeat:**
1. Pick $i_k \in [m]$ w.p. $p_i \propto \|A^{(i)}\|^2$
2. Pick $j_k \in [n]$ w.p. $p^{(j)} \propto \|A^{(j)}\|^2$
3. $z^{(k+1)} = (I - P_{j_k})z^{(k)}$
4. $x^{(k+1)} = x^{(k)} + \frac{y_{i_k} - z_{i_k}^{(k)}}{\|A^{(i_k)}\|^2} A^{(i_k)}$
5. $k = k + 1$

**Exponential convergence** [Z. Freris12]

$$
\mathbb{E} \left\| x^{(k)} - x_{LS} \right\|^2 \leq \left( 1 - \frac{1}{\kappa_F^2(A)} \right)^{\frac{k}{2}} \left( \left\| x_{LS} \right\|^2 + \frac{2 \|b\|^2}{\sigma_{\text{min}}(A)} \right)
$$

*Inspired by Extended Kaczmars method [Pop99]*
Problem II Grand Finale

Randomized Extended Kaczmarz

Initialize: $x^{(0)} = 0; z^{(0)} = y$

Repeat:

Pick node $j_k$ w.p. $d_{j_k}$

$z^{(k+1)} = (I - P_{j_k})z^{(k)}$

Pick $e = (i, j) \in E$ uniformly

$x^{(k+1)} = x^{(k)} + \frac{y_e - z_e^{(k)} - \langle B(e), x^{(k)} \rangle}{2} B(e)$

$k = k + 1$

Same rate of convergence

Gossip Edge-Vertex Solver

Every node $u$: $x_u = 0$, $z_{(u,v)} = y_{(u,v)}$ in $N_u$

Repeat:

Node $u$ activates w.p. $d_u$ & selects random neighbor $v$

- Sends $x_u$ & receives $x_v$

- Performs:

\[
    x_u \leftarrow (x_u + x_v + y_{(u,v)} - z_{(u,v)})/2
\]

Similarly node $v$

- Update weights on edges adjacent to $u$; broadcast to neighbors
Laplacian System

\[ L \mathbf{x} = \mathbf{b} \]

Two solutions:
1. Use REK as before
2. Use RK & gossip averaging to project \( \mathbf{b} \) onto \( \mathbf{1}^\perp \), \( \mathbf{b}' = \mathbf{b} - b_{\text{avg}} \mathbf{1} \)
Summary

- Gossip model of computation
- Randomized Iterative Solvers: RK & REK
- Interplay between randomized solvers & gossip algorithms

Topics not covered:
- Randomized coordinate descent [LL08,Nest10]
- Termination, numerical issues; communication errors, etc
Thank you