MANTRA: Temporal Betweenness Centrality Approximation through Sampling

Antonio Cruciani

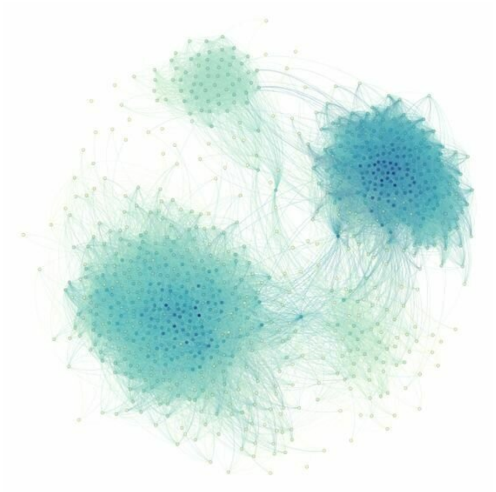
antonio.cruciani@gssi.it

GRAN SASSO SCIENCE INSTITUTE

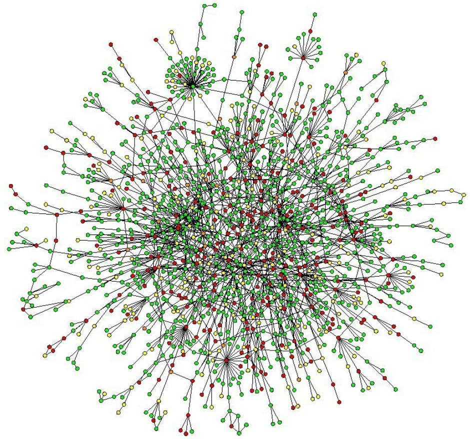
Temporal Networks

Graphs are ubiquitous

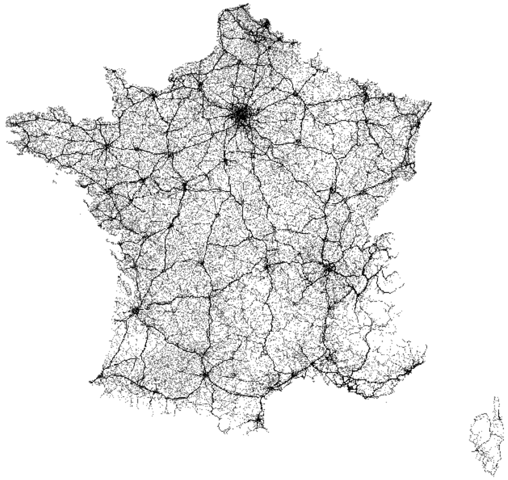
....



Social



Biological

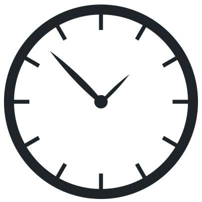


Transport

....

Many of them share a common feature:

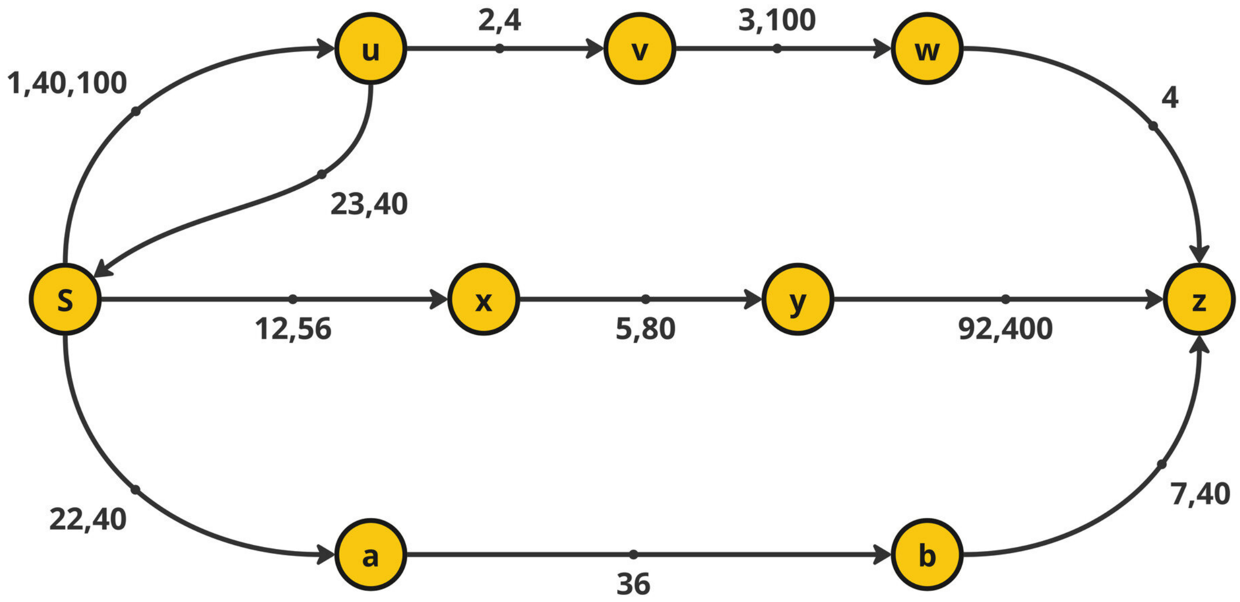
they evolve over time



Temporal Networks

A temporal graph is an ordered triple $\mathcal{G} = (V, \mathcal{E}, T)$, where:

- V is the set of nodes
- $\mathcal{E} = \{(u, v, t) : u, v \in V \wedge t \in [T]\}$ is the set of temporal edges
- $T = \{1, 2, \ldots, |T|\}$ is a set of time steps

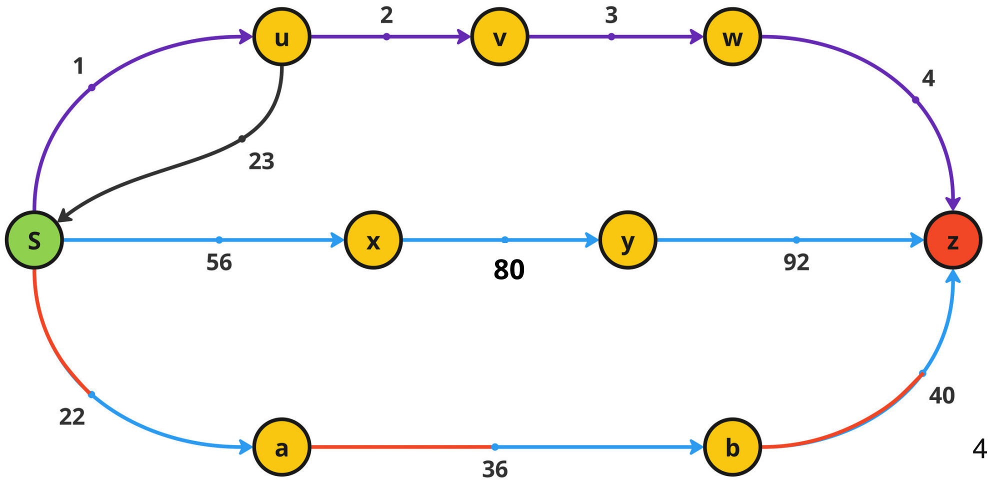


Temporal Betweenness [Buß et al (KDD 2020)]

The temporal betweeness centrality of each node $v \in V$ is defined as

$$b_v^{(\star)} = \frac{1}{n(n-1)} \sum_{\substack{s,z \in V \\ s \neq z \neq v}} \frac{\sigma_{s,z}^{(\star)}(v)}{\sigma_{s,z}^{(\star)}} \in [0,1]$$

- $\sigma_{s,z}^{(\star)}(v)$ is the number of $(\star)$-temporal paths between s and z passing through v
- $\sigma_{s,z}^{(\star)}$ overall number of $(\star)$-temporal paths between s and z
- $(\star)$ can be :
  - **Shortest**
  - **Shortest-Foremost**
  - **Prefix-Foremost**



Temporal Betweenness [Buß et al (KDD 2020)]

The temporal betweeness centrality of each node $v \in V$ is defined as

$$b_v^{(\star)} = \frac{1}{n(n-1)} \sum_{\substack{s,z \in V \\ s \neq z \neq v}} \frac{\sigma_{s,z}^{(\star)}(v)}{\sigma_{s,z}^{(\star)}} \in [0,1]$$

- $\sigma_{s,z}^{(\star)}(v)$ is the number of $(\star)$-temporal paths between s and z passing through v
- $\sigma_{s,z}^{(\star)}$ overall number of $(\star)$-temporal paths between s and z
- $(\star)$ can be :

Path Optimality	Time	Space
Shortest	$\mathcal{O}\left(n^3\|T\|^2\right)$	$\mathcal{O}\left(n\|T\| + \|\mathcal{E}\|\right)$
Shortest-Foremost		
Prefix-Foremost	$\mathcal{O}\left(n\|\mathcal{E}\| \log \|\mathcal{E}\|\right)$	$\mathcal{O}\left(n + \|\mathcal{E}\|\right)$
Foremost	#P-Hard	
Fastest		

Approximating the Temporal Betweenness

- Methods are based on random sampling to estimate the temporal betweenness centrality with an acceptable accuracy

- Problem definition:
  - given $\varepsilon, \delta \in (0,1)$ compute (ε, δ)-approximation set $\tilde{\mathcal{B}}^{(\star)} = \{\tilde{b}_v^{(\star)} : v \in V\}$ such that

$$\mathbf{Pr}\left(\sup_{v \in V} \left| b_v^{(\star)} - \tilde{b}_v^{(\star)} \right| \leq \varepsilon \right) \geq 1 - \delta$$

Approximating the Temporal Betweenness

Works	Temporal Paths	Sample Complexity	Poly. Time	Linear Space	Progressive Sampling	Analysis Techniques
[Santoro and Sarpe 2022]	Shortest	?	Yes	No	No	Empirical Bernstein Bound
This work	All•	Yes	Yes	Yes	Yes	VC-Dim., Rademacher Avg. & Variance Aware bounds

• Temporal paths that can be computed in polynomial time.

How does MANTRA work?

MANTRA in three lines

- MANTRA quickly "observes" the temporal graph.
- MANTRA starts sampling, computing the approximation as it goes.
- At predefined intervals, MANTRA checks **two stopping conditions** to understand, using the sample, whether the current approximation has the desired quality.

Supremum Deviation

Given a set of functions $\mathcal{F}$ from a domain $\mathcal{D}$ and a sample $\mathcal{S}$

$$a_f(\mathcal{S}) = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|} f(s_i) \qquad \mu_f = \mathbb{E}[a_f(\mathcal{S})]$$

The Supremum Deviation

$$\mathbf{SD}(\mathcal{F},\mathcal{S}) = \sup_{f\in\mathcal{F}} |a_f(\mathcal{S}) - \mu_f|$$

Goal: To find an Upper bound for $\mathbf{SD}(\mathcal{F},\mathcal{S})$

Rademacher Averages

c-Monte Carlo Emprical Rademacher Averages (c-MCERA)

$$R_r^c(\mathcal{F}, \mathcal{S}, \boldsymbol{\lambda}) = \frac{1}{c} \sum_{j=1}^{c} \sup_{f \in \mathcal{F}} \frac{1}{r} \sum_{s_i \in \mathcal{S}} \lambda_{j,i} f(s_i)$$

$\boldsymbol{\lambda} \in \{-1, +1\}^{c \times |\mathcal{S}|}, r = |\mathcal{S}|$

Variance Dependent Bound

$$\tilde{R} = R_r^c(\mathcal{F}, \mathcal{S}, \boldsymbol{\lambda}) + \sqrt{\frac{4\mathcal{W}_{\mathcal{F}}(\mathcal{S}) \ln(4/\delta)}{cr}}$$

$$R = \tilde{R} + \frac{\ln(4/\delta)}{r} + \sqrt{\left(\frac{\ln(4/\delta)}{r}\right)^2 + \frac{2\ln(4/\delta)\tilde{R}}{r}}$$

$$\xi = 2R + \sqrt{\frac{2\ln(4/\delta)(\hat{v}+4R)}{r}} + \frac{\ln(4/\delta)}{3r}$$

$\mathbf{SD}(\mathcal{F}, \mathcal{S}) \leq \xi$ With Probability at least $1 - \delta$

Empirical Wimpy Variance

$$\mathcal{W}_{\mathcal{F}}(\mathcal{S}) = \sup_{f \in \mathcal{F}} \frac{1}{r} \sum_{s_i \in \mathcal{S}} (f(s_i))^2$$

Back to the Temporal Betweenness

Functions $$f_v(s,z) = \frac{\sigma_{s,z}^{(\star)}(v)}{\sigma_{s,z}^{(\star)}}$$

Domain $$\mathcal{D} = \{(s,z) \in V \times V : s \neq z\}$$

Sample Mean $$\tilde{b}_v^{(\star)} = \frac{1}{|\mathcal{S}|} \sum_{(s,z)\in\mathcal{S}} f_v(s,z)$$

Emp. Wimpy Variance $$\mathcal{W}_{\mathcal{F}}(\mathcal{S}) = \sup_{v\in V} \frac{1}{|\mathcal{S}|} \sum_{(s,z)\in\mathcal{S}} (f_v(s,z))^2$$

Sample Size for (ε, δ)-approximation

Given $\varepsilon, \delta \in (0,1)$

Vapnik–Chervonenkis (VC) Dimension

$$|\mathcal{S}| = \frac{0.5}{\varepsilon^2}\left(\lfloor \log D^{(\star)} - 2 \rfloor + 1 + \ln\left(\frac{1}{\delta}\right)\right)$$

↑

Maximum number of nodes in the $(\star)$-temporal optimal path

$\mathbf{SD}(\mathcal{F}, \mathcal{S}) \leq \varepsilon$ with probability $\geq 1 - \delta$

Sample Size for (ε, δ)-approximation

Given $\varepsilon, \delta \in (0, 1)$

Variance-Aware

$$|\mathcal{S}| \in \mathcal{O}\left(\frac{\hat{v} + \varepsilon}{\varepsilon^2} \ln\left(\frac{\rho^{(\star)}}{\delta \hat{v}}\right)\right)$$

$$\max_{v \in V} \mathbf{Var}(\tilde{b}_v^{(\star)}) \leq \hat{v}$$

$$\frac{1}{n(n-1)} \sum_{s,z \in V} |\mathbf{Int}(tp_{sz})|$$

$\mathbf{SD}(\mathcal{F}, \mathcal{S}) \leq \varepsilon$ with probability $\geq 1 - \delta$

Fast Approximation of $D^{(\star)}$ and $\rho^{(\star)}$

Very high-level idea

Perform k $(\star)$-Temporal BFS from random nodes

With a sample of $k = \Theta\left(\frac{\ln n}{\varepsilon^2}\right)$ nodes, we can approximate w.h.p.

- $\rho^{(\star)}$ with the absolute error bounded by $\varepsilon \frac{D^{(\star)}}{\zeta}$
- $D^{(\star)}$ with the absolute error bounded by $\frac{\varepsilon}{\zeta}$
- ζ with the absolute error bounded by ε

Temporal Connectivity Rate

$$\zeta = \frac{1}{n(n-1)} \sum_{\substack{u,v \in V \\ u \neq v}} \mathbf{1}[u \rightsquigarrow v] \in [0,1]$$

MANTRA

Algorithm 1: MANTRA

Data: Temporal graph $\mathcal{G}$, $(\star)$ temporal path optimality, precision $\varepsilon \in (0,1)$, failure probability $\delta \in (0,1)$, bootstrap iterations k, and number of Monte Carlo trials c.

Result: Absolute ε-approximation of the $(\star)$-tbc w.p. of at least $1-\delta$.

$\mathcal{B}, \mathcal{W} = [0, \ldots, 0] \in \mathbb{R}^n$ `// tbc and wimpy variance arrays`
$\mathcal{S}_0 = \{\emptyset\}$
$i = 0$
$\xi = 1$
$\mathcal{D} = \{(s,z) \in V \times V \wedge s \neq z\}$
`// Bootstrap Phase`
$\rho^{(\star)}, \hat{v} =$ `Run` $(\star)$`-Temporal BFS`$(k, \mathcal{D})$
$\omega =$ `DrawSufficientSampleSize()`
$\{s_i\}_{i\geq 1} =$ `SamplingSchedule`$(\omega, \hat{v}, \delta)$
`// Estimation Phase`
while *true* **do**
    $i = i + 1$;
    $h = s_i - s_{i-1}$
    $\mathcal{X} = \{\emptyset\}$
    for $i = 1$ ***to*** h **do**
        $(s,z) =$ `uniform_random_sample`$(\mathcal{D})$
        `Compute` $(\star)$`-Temporal Paths`(s,z)
        $\mathcal{X} = \mathcal{X} \cup \{(s,z)\}$
    $\mathcal{S}_i = \mathcal{S}_{i-1} \cup \{\mathcal{X}\}$
    `Calculate` $\tilde{\mathcal{B}}$ `and` $\mathcal{W}$ `from` $\mathcal{S}_i$
    $\xi =$ `ComputeSDBound`$(\tilde{\mathcal{B}}, \mathcal{W}, \delta/2^i, s_i, h, c)$
    if $|\mathcal{S}_i| \geq \omega$ ***or*** $\xi \leq \varepsilon$ **then return** $\{(1/|\mathcal{S}_i|) \cdot \mathcal{B}[u] : u \in V\}$

Bootstrap Phase

Estimation Phase

Experimental Setting

All the algorithms have been implemented in Julia

Parameters:

$$\varepsilon \in \{0.1, 0.07, 0.05, 0.01, 0.007.0.005, 0.001\}$$

$$\delta = 0.1$$

$$c = 25$$

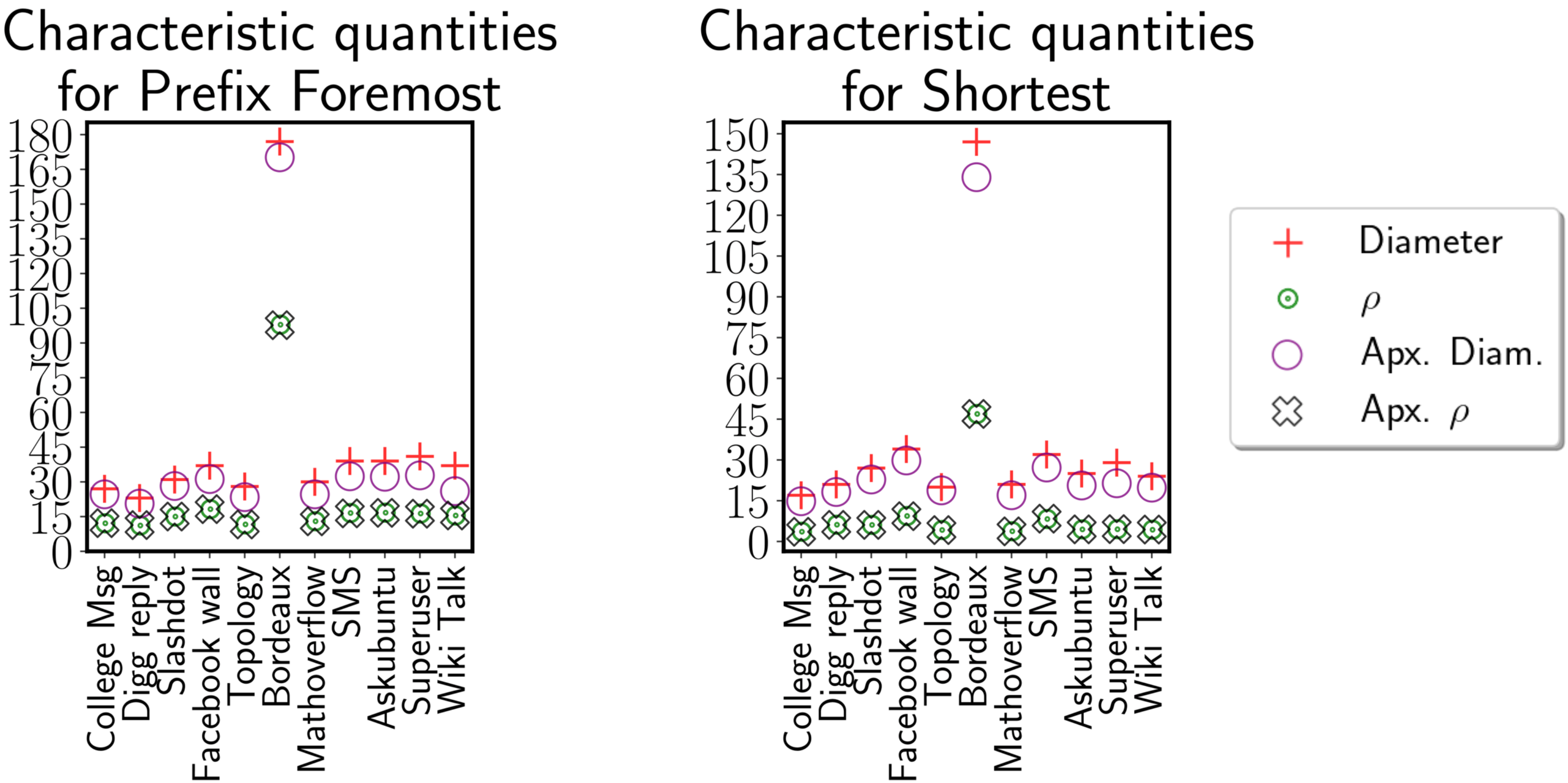
Every algorithm is run 10 times

Machine used: Server with Intel Xeon Gold 6248R (3.0GHz)
32 cores and 1TB RAM

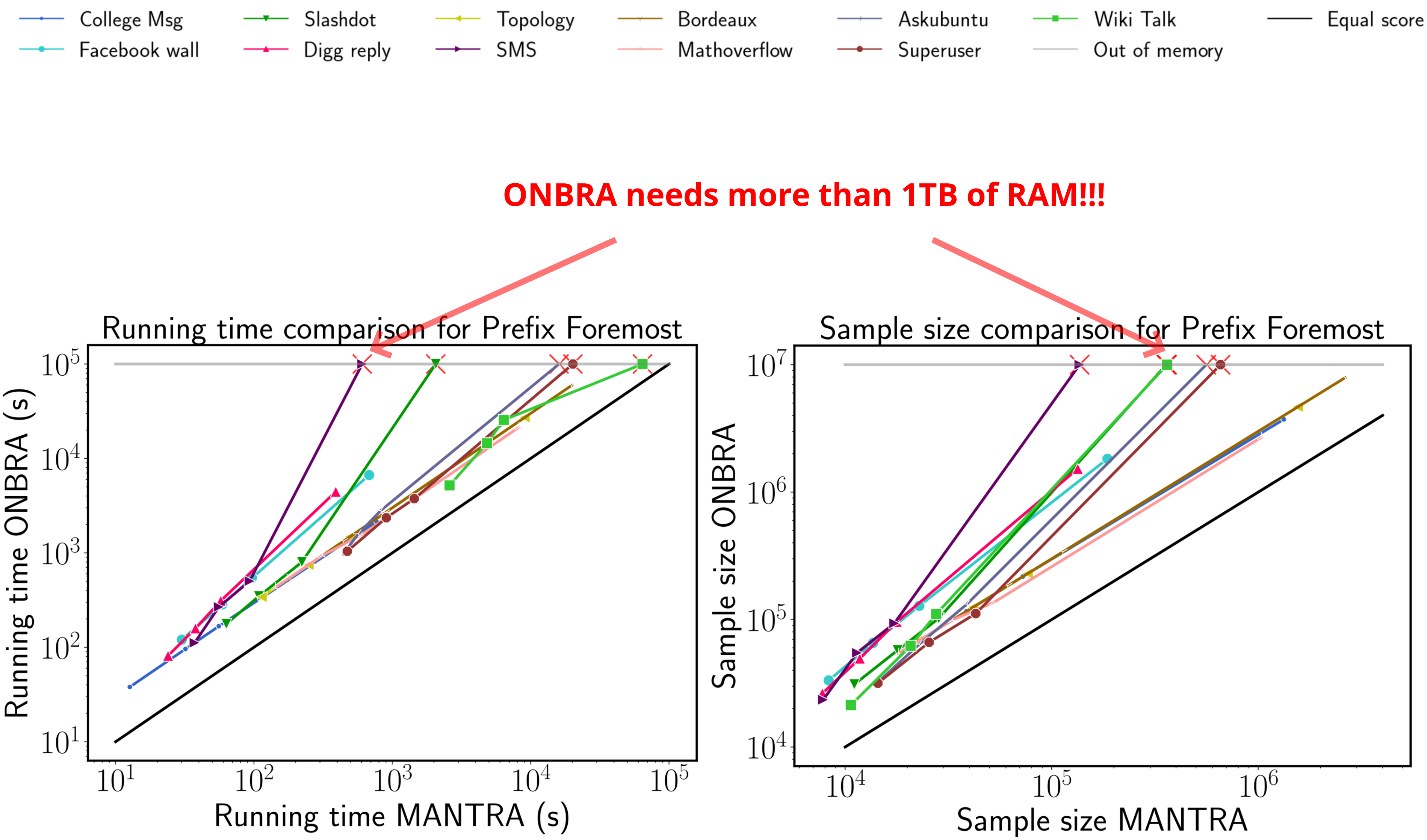
Temporal Networks

Data set	n	$\|\mathcal{E}\|$	$\|T\|$	ζ	$b_{\mathtt{max}}^{(\mathtt{pfm})}$	$b_{\mathtt{max}}^{(\mathtt{sh})}$	$b_{\mathtt{max}}^{(\mathtt{sfm})}$	Type
College msg	1899	59798	58911	0.5	0.0718	0.0319	0.0365	D
Digg reply	30360	86203	82641	0.02	0.0019	0.0015	0.0016	D
Slashdot	51083	139789	89862	0.07	0.0128	0.0074	0.0085	D
Facebook Wall	35817	198028	194904	0.04	0.0034	0.0024	0.0028	D
Topology	16564	198038	32823	0.53	0.0921	0.0654	0.0681	U
Bordeaux•	3435	236075	60582	0.84	0.1210	0.1383	0.1269	D
Mathoverflow	24759	390414	389952	0.33	0.0522	0.0282	0.0287	D
SMS	44090	544607	467838	0.008	0.0019	0.0010	0.0012	D
Askubuntu	157222	726639	724715	0.169	0.0214	0.0156	0.0154	D
Super user	192409	1108716	1105102	0.21	0.0261	0.0165	0.0182	D
Wiki Talk	1094018	6092445	5799206	0.069	0.0089	0.0155	0.0153	D

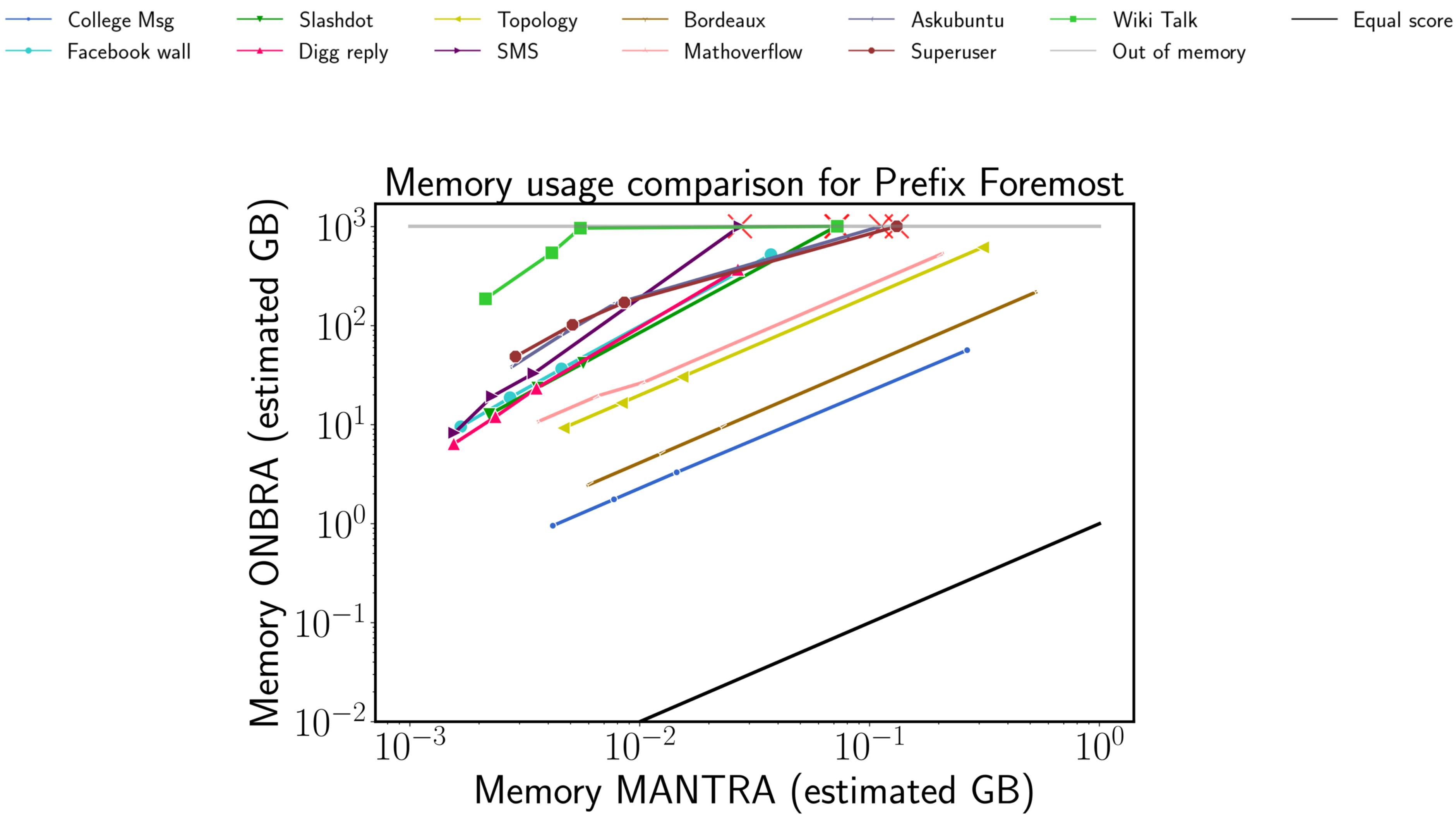
Temporal Networks: properties



Speed an Sample size MANTRA vs ONBRA



Space MANTRA vs ONBRA



Conclusions

- Introduced MANTRA, a novel sampling-based approximation algorithm for the temporal betweenness centrality
- Provided a sample-complexity analysis for the temporal betweenness estimation problem
- Provided a sampling-based approximation algorithm for temporal distance-based metrics in temporal graphs
- Theoretically and experimentally showed the advantage of using our framework over the state of the art

Future directions

- Use the novel temporal graph traversal proposed by Brunelli et al. in KDD 2024 to speed-up MANTRA

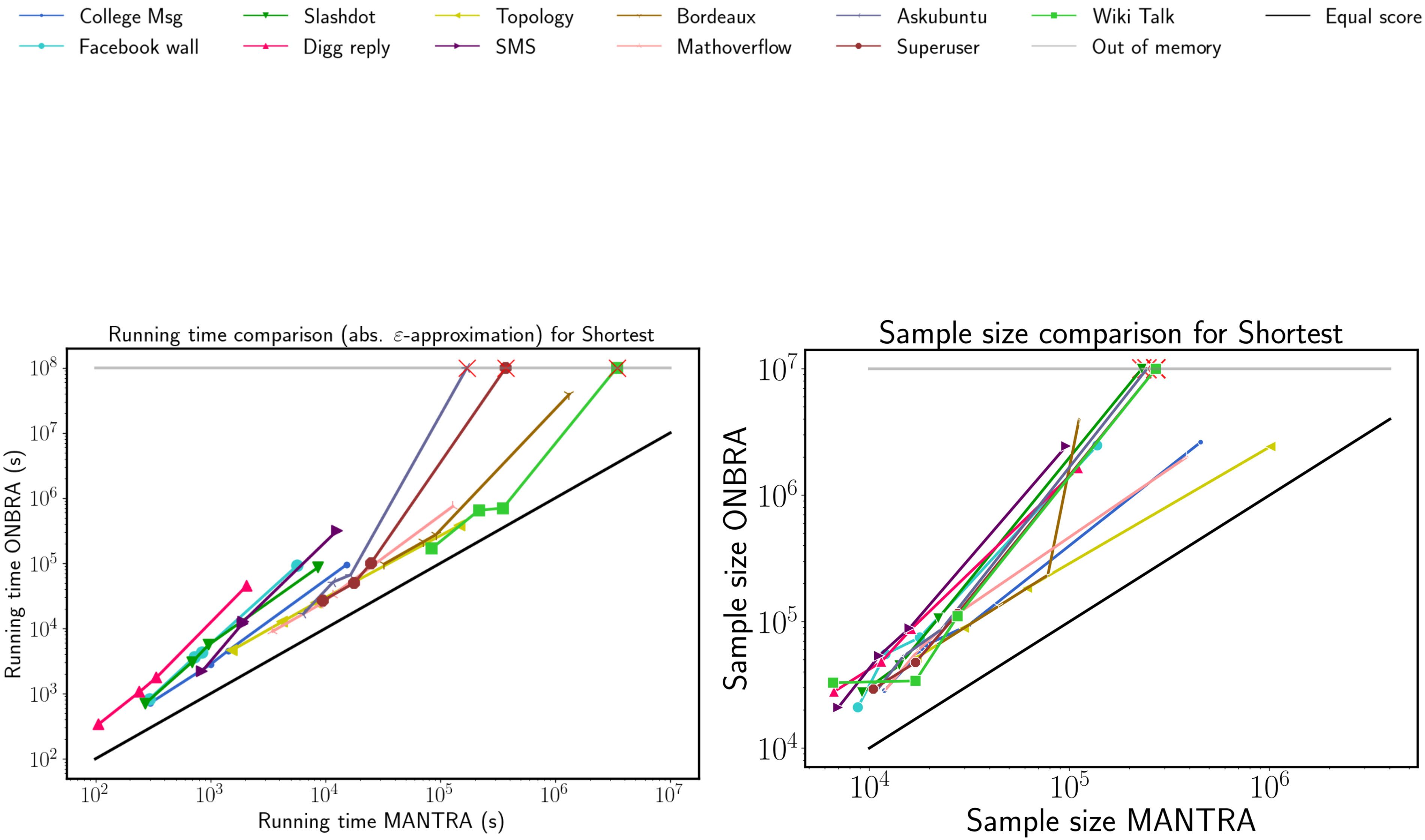
- Extend MANTRA to the computation of set centralities as for the static case [Pellegrina KDD 2023]

- Use MANTRA to find communities in temporal graphs

Thank You!

MANTRA

MANTRA vs ONBRA For shortest TBC



MANTRA vs ONBRA For shortest TBC