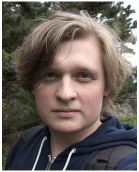
A Distance Measure for the Analysis of Polar Opinion Dynamics in Social Networks

Victor Amelkin

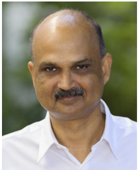
University of California, Santa Barbara
Department of Computer Science
victor@cs.ucsb.edu

Contributors[1,2]



Victor Amelkin
UC Santa Barbara
victor@cs.ucsb.edu

Petko Bogdanov
University at Albany, SUNY
pbogdanov@albany.edu



Ambuj K. Singh
UC Santa Barbara
ambuj@cs.ucsb.edu

[1] Victor Amelkin, Petko Bogdanov, and Ambuj K Singh. "A Distance Measure for the Analysis of Polar Opinion Dynamics in Social Networks". In: *Proc. IEEE ICDE*. 2017, pp. 159–162.

[2] Victor Amelkin, Petko Bogdanov, and Ambuj K. Singh. "A Distance Measure for the Analysis of Polar Opinion Dynamics in Social Networks (Extended Paper)". In: *arXiv:1510.05058 [cs.SI]* (2015).

Table of Contents

Introduction

- Directed social network, $|V| = n$ users, $|E| = m$ social ties
- Network is sparse: $m = \mathcal{O}(n)$
- User opinions are **polar** (e.g., *the Republicans vs. the Democrats*)
- Opinion $\in \{+1, 0, -1\}$
- Network structure does not change much, but user opinions evolve

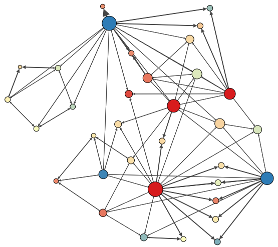
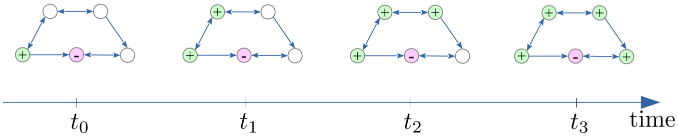


Figure: *Zachary's Karate Club* network[3]

[3] Wayne Zachary. "An information flow model for conflict and fission in small groups". In: *Journal of Anthropological Research* (1977), pp. 452–473.

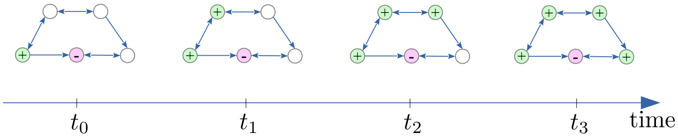
Polar Opinion Dynamics

- **Network state** $G_t \in \{+1, 0, -1\}^n$: opinions of all users at time t
- A time series of network states



Polar Opinion Dynamics

- **Network state** $G_t \in \{+1, 0, -1\}^n$: opinions of all users at time t
- A time series of network states

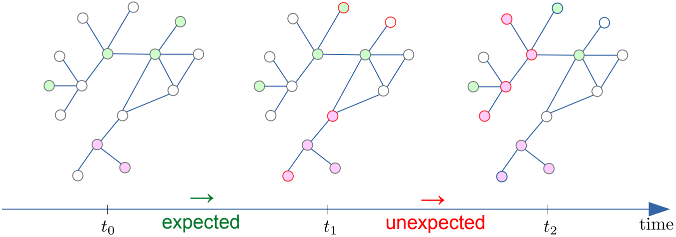


Questions:

- How does the network evolve?
- What will be the future opinions of individual users?
- When does the network "behave" unexpectedly?

Application I: Anomalous Event Detection

- $d_t = d(G_t, G_{t+1})$: *"the amount of change"* in the network's state
- d_t measures the *unexpectedness* of transition $G_t \rightarrow G_{t+1}$
- What is *expected* is determined by a given opinion dynamics model

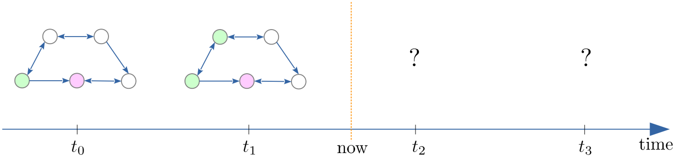


- Anomaly: an unexpected value in the series $d_0, d_1, d_2, \ldots, d_t$
- A distance-based approach to anomaly detection[4]

[4] Stephen Ranshous et al. "Anomaly detection in dynamic networks: a survey". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 7.3 (2015), pp. 223–247.

Application II: User Opinion Prediction

- $d_t = d(G_t, G_{t+1})$ – *"the amount of change"* in the network's state
- d_t measures the *unexpectedness* of transition $G_t \to G_{t+1}$
- What is *expected* is determined by a given opinion dynamics model
- Having observed the network state's evolution $G_0, G_1, \ldots, G_{now}$



we would like to predict G_{future}

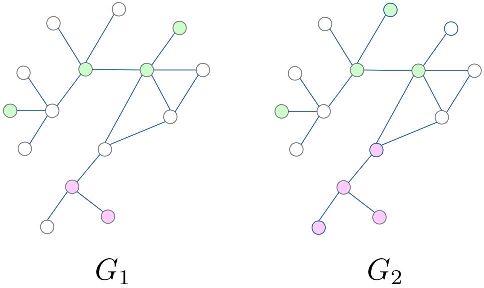
- Distance-based approach to future network state prediction:

$$d_0, d_1, \ldots, d_{now} \xrightarrow{extrapolate} d_{future} \xrightarrow{reconstruct} G_{future}$$

Distance Measure-Based Analysis

- Central question:

  How to measure **the distance** $d(G_1, G_2)$ **between network states**?



- The distance measure $d(\bullet, \bullet)$ should
  - ▷ capture how polar opinions evolve in the network;
  - ▷ be efficiently computable;
  - ▷ be a metric.

Existing Vector Space Distance Measures

- Coordinate-wise comparison
  - ℓ_p — $d(x,y) = \left(\sum_i |x_i - y_i|^p\right)^{1/p}$
  - Hamming — $d(x,y) = \sum_i \delta_{x_i,y_i}$
  - Canberra — $d(x,y) = \sum_i \frac{|x_i - y_i|}{|x_i| + |y_i|}$
  - Jaccard — $d(x,y) = \frac{|x \cap y|}{|x \cup y|}$
  - Cosine — $d(x,y) = \cos\widehat{(x,y)} = \frac{\langle x,y \rangle}{\|x\|\ \|y\|}$
  - Kullback-Leibler — $d(x,y) = (d_{KL}(x||y)) = \sum_i \ln\left[x_i/y_i\right] x_i$
- Using the difference vector
  - Quadratic Form — $d(x,y) = \sqrt{(x-y)^T A (x-y)}$
  - Mahalanobis — $d(x,y) = \sqrt{(x-y)^T cov^{-1}(x,y)(x-y)}$

Existing Network-Specific Distance Measures

- Isomorphism-based distance measures[5]
- Graph Edit Distance[6]
- Iterative distance measures[7]
- Graph Kernels[8]
- Feature-based distance measures[9]

---

[5] Horst Bunke and Kim Shearer. "A graph distance metric based on the maximal common subgraph". In: *Pattern recognition letters* 19.3 (1998), pp. 255–259.

[6] Xinbo Gao et al. "A survey of Graph Edit Distance". In: *Pattern Analysis and Applications* 13.1 (2010), pp. 113–129.

[7] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. "Similarity flooding: A versatile graph matching algorithm and its application to schema matching". In: *IEEE Data Engineering*. 2002, pp. 117–128.

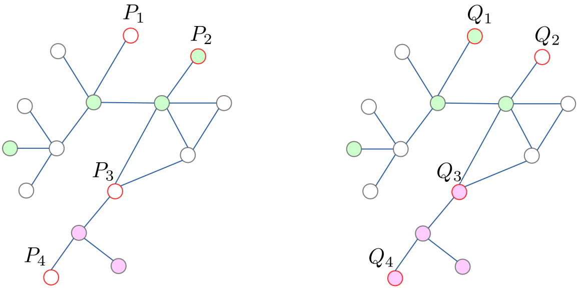
[8] S Vichy N Vishwanathan et al. "Graph kernels". In: *The Journal of Machine Learning Research* 11 (2010), pp. 1201–1242.

[9] Owen Macindoe and Whitman Richards. "Graph comparison using fine structure analysis". In: *IEEE SocialCom*. IEEE. 2010, pp. 193–200.

Existing Network-Specific Distance Measures

- Isomorphism-based distance measures
  - compare networks structurally
  - disregard node states
- Graph Edit Distance
  - edit distance over node/edge insertion, deletion, substitution operations
  - mostly, structure-driven; expensive to compute
- Iterative distance measures
  - nodes are similar if their neighborhoods are similar
  - hard to account for node state differences in a socially meaningful way; expensive to compute
- Graph Kernels
  - compare substructures—walks, paths, cycles, trees—of non-aligned (small) networks
  - opinion dynamics-unaware; expensive to compute
- Feature-based distance measures
  - compare degree, clust. coeff., betweenness, diameter, frequent substructures, spectra
  - only look at summaries; does not capture opinion dynamics

Social Network Distance (SND): Overview[5]



P $\qquad\qquad$ Q

$$\mathrm{SND}(P,Q) \approx -\log \mathbb{P}\left\{\begin{array}{ll} P_1 \bigcirc \rightsquigarrow \bigcirc Q_1, & P_3 \bigcirc \rightsquigarrow \bigcirc Q_3, \\ P_2 \bigcirc \rightsquigarrow \bigcirc Q_2, & P_4 \bigcirc \rightsquigarrow \bigcirc Q_4. \end{array}\right\}$$

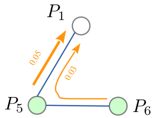
[5]Amelkin, Bogdanov, and Singh, "A Distance Measure for the Analysis of Polar Opinion Dynamics in Social Networks (Extended Paper)".

Social Network Distance (SND): Overview

- Exact computation of $\mathbb{P}$: computationally hard
- Assume user activations are independent

$$\mathbb{P}\left\{ P_1 \rightsquigarrow Q_1 \;\middle|\; P_3 \rightsquigarrow Q_3 \right\} = \mathbb{P}\left\{ P_1 \rightsquigarrow Q_1 \right\}$$

- Assume activations happens via the most likely scenarios



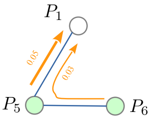
$$\mathbb{P}\left\{ P_5 \xrightarrow{infects} P_1 \right\} = 0.05 > 0.03 = \mathbb{P}\left\{ P_6 \xrightarrow{infects} P_1 \right\}$$

Social Network Distance (SND): Overview

- Exact computation of $\mathbb{P}$: computationally hard
- Assume user activations are independent
  $\sim$ "opinion flows" in the network do not interfere with each other

$$\mathbb{P}\left\{ P_1 \rightsquigarrow Q_1 \mid P_3 \rightsquigarrow Q_3 \right\} = \mathbb{P}\left\{ P_1 \rightsquigarrow Q_1 \right\}$$

- Assume activations happens via the most likely scenarios
  $\sim$ opinions spread via shortest paths



$$\mathbb{P}\left\{ P_5 \xrightarrow{infects} P_1 \right\} = 0.05 > 0.03 = \mathbb{P}\left\{ P_6 \xrightarrow{infects} P_1 \right\}$$

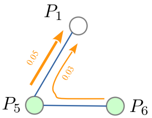
- $\Rightarrow$ SND is defined as **a transportation problem**

Social Network Distance (SND): Overview

- Exact computation of $\mathbb{P}$: computationally hard
- Assume user activations are independent
  $\sim$ "opinion flows" in the network do not interfere with each other

$$\mathbb{P}\left\{ P_1 \bigcirc \rightsquigarrow \bigcirc\ Q_1 \mid P_3 \bigcirc \rightsquigarrow \bigcirc\ Q_3 \right\} = \mathbb{P}\left\{ P_1 \bigcirc \rightsquigarrow \bigcirc\ Q_1 \right\}$$

- Assume activations happens via the most likely scenarios
  $\sim$ opinions spread via shortest paths

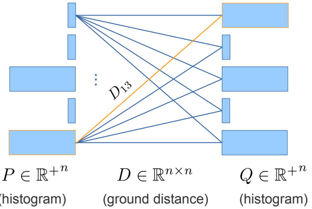


$$\mathbb{P}\left\{ P_5 \xrightarrow{infects} P_1 \right\} = 0.05 > 0.03 = \mathbb{P}\left\{ P_6 \xrightarrow{infects} P_1 \right\}$$

- $\Rightarrow$ SND is defined as **a transportation problem** that can be exactly solved in $\mathcal{O}(n)$ /*under some reasonable assumptions*/

Earth Mover's Distance (EMD) as a Basic Primitive

- Earth Mover's Distance (EMD): "edit distance for histograms"
- Edit: transportation of a mass unit from i'th to j'th bin at cost D_{ij}



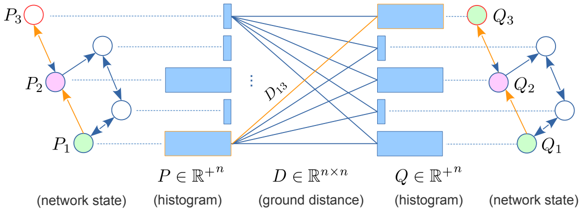
$$\mathrm{EMD}(P, Q, D) = \sum_{i,j=1}^{n} D_{ij} \widehat{f}_{ij} \Big/ \sum_{i,j=1}^{n} \widehat{f}_{ij},$$

$$\sum_{i,j=1}^{n} f_{ij} D_{ij} \to \min, \quad \sum_{i,j=1}^{n} f_{ij} = \min \left\{ \sum_{i=1}^{n} P_i, \sum_{i=1}^{n} Q_i \right\}$$

$$f_{ij} \geq 0, \sum_{j=1}^{n} f_{ij} \leq P_i, \sum_{i=1}^{n} f_{ij} \leq Q_j, (1 \leq i, j \leq n)$$

Earth Mover's Distance (EMD) as a Basic Primitive

- Earth Mover's Distance (EMD): "edit distance for histograms"
- Edit: transportation of a mass unit from i'th to j'th bin at cost D_{ij}

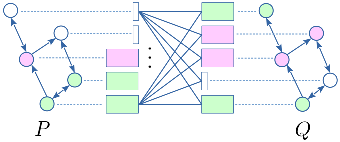


$$\mathrm{EMD}(P, Q, D) = \sum_{i,j=1}^{n} D_{ij} \widehat{f}_{ij} \Big/ \sum_{i,j=1}^{n} \widehat{f}_{ij},$$

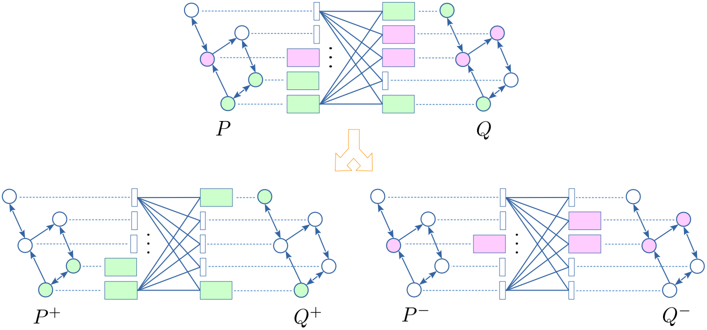
$$\sum_{i,j=1}^{n} f_{ij} D_{ij} \to \min, \quad \sum_{i,j=1}^{n} f_{ij} = \min\left\{\sum_{i=1}^{n} P_i, \sum_{i=1}^{n} Q_i\right\}$$

$$f_{ij} \geq 0, \sum_{j=1}^{n} f_{ij} \leq P_i, \sum_{i=1}^{n} f_{ij} \leq Q_j, (1 \leq i, j \leq n)$$

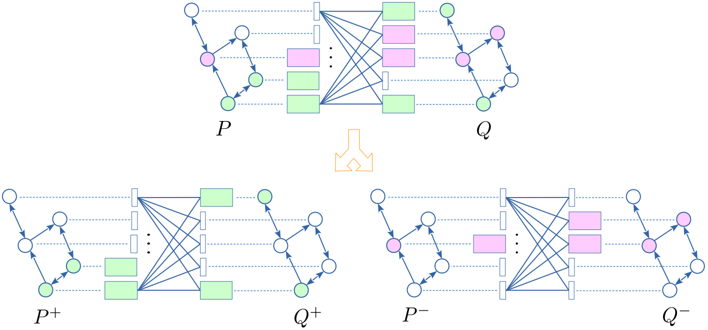
Social Network Distance (SND) – Definition



Social Network Distance (SND) – Definition

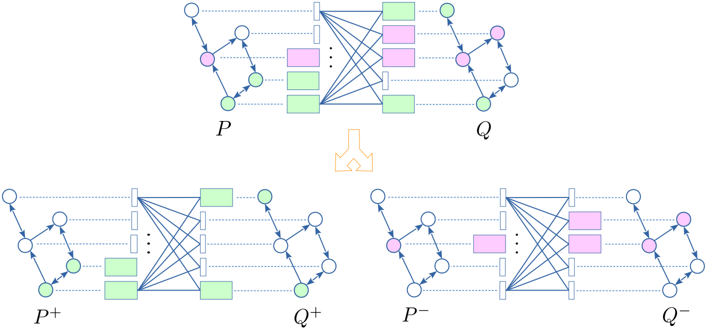


Social Network Distance (SND) – Definition



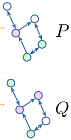
$$\mathrm{SND}(P,Q) = \begin{array}{l} \mathrm{EMD}(P^+, Q^+, D(P,+)) + \mathrm{EMD}(P^-, Q^-, D(P,-)) + \\ \\ \mathrm{EMD}(Q^+, P^+, D(Q,+)) + \mathrm{EMD}(Q^-, P^-, D(Q,-)) \end{array}$$

Social Network Distance (SND) – Definition



Ground distance computed in:

$$\mathrm{SND}(P,Q) = \mathrm{EMD}(P^+, Q^+, D(P,+)) + \mathrm{EMD}(P^-, Q^-, D(P,-)) + \mathrm{EMD}(Q^+, P^+, D(Q,+)) + \mathrm{EMD}(Q^-, P^-, D(Q,-))$$



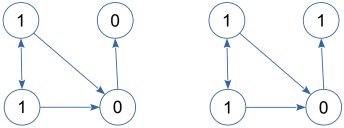
Opinion type "transported": + –

EMD*– Redesign of Earth Mover's Distance for SND

- EMD has 2 problems:
  - (i) cannot adequately compare histograms with different total mass
  - (ii) cannot express a single user infecting *multiple* other users

- EMD*—generalization of EMD—resolves both issues.

P Q

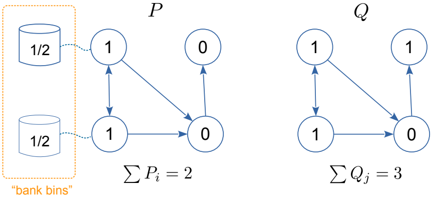


$\sum P_i = 2$ $\sum Q_j = 3$

EMD*– Redesign of Earth Mover's Distance for SND

- EMD has 2 problems:
  - (i) cannot adequately compare histograms with different total mass
  - (ii) cannot express a single user infecting *multiple* other users

- EMD*—generalization of EMD—resolves both issues.



  - (i) mass mismatch penalty is related to the network's structure
  - (ii) users can spend "extra mass" to infect more neighbors

$\mathrm{EMD}^{\star}$– Redesign of Earth Mover's Distance for SND

- EMD has 2 problems:
  - (i) cannot adequately compare histograms with different total mass
  - (ii) cannot express a single user infecting *multiple* other users

- $\mathrm{EMD}^{\star}$—generalization of EMD—resolves both issues.

$$\mathrm{EMD}^{\star}(P,Q) = \mathrm{EMD}(\widetilde{P},\widetilde{Q},\widetilde{D}) \max\left\{\sum P_i, \sum Q_j\right\},$$

$$\widetilde{P} = \left[P, P^{(1)}, \dots, P^{(n)}\right], \quad \widetilde{Q} = \left[Q, Q^{(1)}, \dots, Q^{(n)}\right],$$

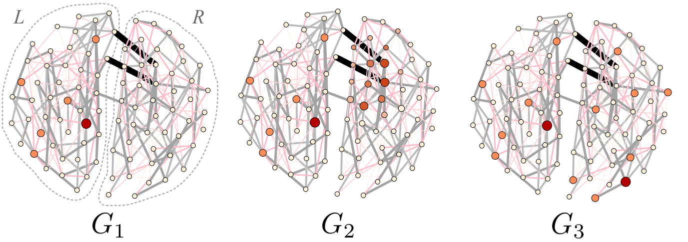
$$\widetilde{D} = \left[\begin{array}{c|c} D & D + \mathbb{1}_n \otimes \gamma^T \\ \hline D + \mathbb{1}_n^T \otimes \gamma & D + \mathbb{1}_n \otimes \gamma^T + \mathbb{1}_n^T \otimes \gamma - 2\,\mathrm{diag}(\gamma) \end{array}\right],$$

$$P^{(i)} = \begin{cases} P_i \;/\; \left(\sum_{j=1}^{n} Q_j - \sum_{k=1}^{n} P_k\right), & \text{if } \sum Q_j > \sum P_k, \\ 0, & \text{otherwise.} \end{cases}$$

$P^{(i)}$: capacity of the i'th bank bin,

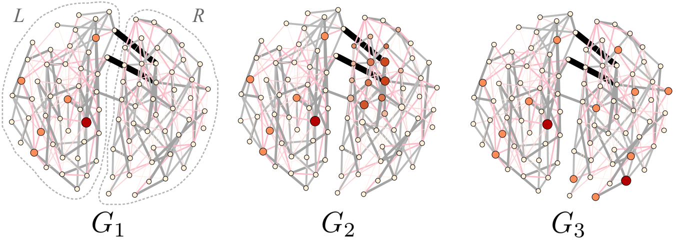
$\gamma = [\gamma_1, \dots, \gamma_n]^{\mathsf{T}}$: ground distances to/from bank bins.

EMD* vs. EMD



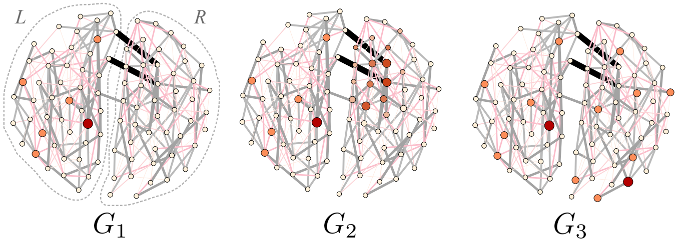
- Mass distribution in cluster L is identical in all G_1, G_2, G_3
- $G_1 \to G_2$: mass propagates from L to R through "the bridges"
- $G_1 \to G_3$: same amount of mass randomly distributed over R

$\mathrm{EMD}^{\star}$ vs. EMD



- Mass distribution in cluster L is identical in all G_1, G_2, G_3
- $G_1 \to G_2$: mass propagates from L to R through "the bridges"
- $G_1 \to G_3$: same amount of mass randomly distributed over R
- Expected: $d(G_1, G_2) < d(G_1, G_3)$

$\mathrm{EMD}^\star$ vs. EMD



- Mass distribution in cluster L is identical in all G_1, G_2, G_3
- $G_1 \to G_2$: mass propagates from L to R through "the bridges"
- $G_1 \to G_3$: same amount of mass randomly distributed over R
- Expected: $d(G_1, G_2) < d(G_1, G_3)$
- Of all existing versions of EMD, only $\mathrm{EMD}^\star$ captures this intuition

Computation of SND – Overview

- Computing $\mathrm{SND} \sim$ computing 4 instances of $\mathrm{EMD}^\star$

$$\begin{aligned}\mathrm{SND}(P,Q) = &\,\mathrm{EMD}^\star(P^+, Q^+, D(P,+)) + \mathrm{EMD}^\star(P^-, Q^-, D(P,-)) + \\ &\,\mathrm{EMD}^\star(Q^+, P^+, D(Q,+)) + \mathrm{EMD}^\star(Q^-, P^-, D(Q,-)).\end{aligned}$$

- Computation of a single instance of $\mathrm{EMD}^\star$ involves:
  - ▷ computing ground distance D
  - ▷ solving the underlying transportation problem

Computation of SND – Overview

- Computing $\mathrm{SND} \sim$ computing 4 instances of $\mathrm{EMD}^\star$

$$\mathrm{SND}(P,Q) = \mathrm{EMD}^\star(P^+, Q^+, D(P,+)) + \mathrm{EMD}^\star(P^-, Q^-, D(P,-)) + \mathrm{EMD}^\star(Q^+, P^+, D(Q,+)) + \mathrm{EMD}^\star(Q^-, P^-, D(Q,-)).$$

- Computation of a single instance of $\mathrm{EMD}^\star$ involves:
  - ▷ computing ground distance D
  - ▷ solving the underlying transportation problem
- Direct computation:
  - ▷ computing ground distance D

    – all-to-all shortest paths
  - ▷ solving the underlying transportation problem

    – Karmakar's algorithm / transportation simplex

Computation of SND – Overview

- Computing $\mathrm{SND} \sim$ computing 4 instances of $\mathrm{EMD}^\star$

$$\mathrm{SND}(P,Q) = \mathrm{EMD}^\star(P^+, Q^+, D(P,+)) + \mathrm{EMD}^\star(P^-, Q^-, D(P,-)) + \\ \mathrm{EMD}^\star(Q^+, P^+, D(Q,+)) + \mathrm{EMD}^\star(Q^-, P^-, D(Q,-)).$$

- Computation of a single instance of $\mathrm{EMD}^\star$ involves:
  - ▷ computing ground distance D
  - ▷ solving the underlying transportation problem
- Direct computation:
  - ▷ computing ground distance D
    – all-to-all shortest paths $\mathcal{O}(n^2 \log n)$
  - ▷ solving the underlying transportation problem
    – Karmakar's algorithm / transportation simplex "$>$" $\mathcal{O}(n^3)$

Computation of SND – Overview

- Computing $\mathrm{SND} \sim$ computing 4 instances of $\mathrm{EMD}^\star$

$$\begin{aligned}\mathrm{SND}(P,Q) = {} & \mathrm{EMD}^\star(P^+, Q^+, D(P,+)) + \mathrm{EMD}^\star(P^-, Q^-, D(P,-)) + \\ & \mathrm{EMD}^\star(Q^+, P^+, D(Q,+)) + \mathrm{EMD}^\star(Q^-, P^-, D(Q,-)).\end{aligned}$$

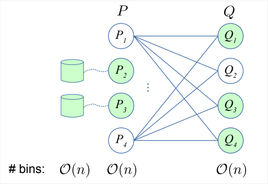
- Computation of a single instance of $\mathrm{EMD}^\star$ involves:
  - ▷ computing ground distance D
  - ▷ solving the underlying transportation problem
- Direct computation:
  - ▷ computing ground distance D

    – all-to-all shortest paths $\mathcal{O}(n^2 \log n)$
  - ▷ solving the underlying transportation problem

    – Karmakar's algorithm / transportation simplex ">" $\mathcal{O}(n^3)$
- Solution: exploit the problem's structure; use specialized algorithms

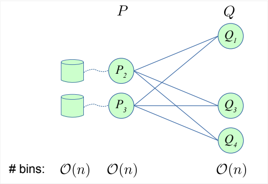
Efficient Computation of SND / $\mathrm{EMD}^\star$

- Challenge: efficiently compute $\mathrm{EMD}^\star(P, Q, D)$ over sparse network
  - D (all-to-all shortest paths): $\mathcal{O}(n^2 \log n)$
  - $\mathrm{EMD}^\star$ (BP min-cost flow): $\mathcal{O}(n^3 \log n)$
- Assumption 1: number n_Δ of users who changed their opinions $\ll n$
- Assumption 2: $D_{ij} \in \mathbb{Z}^+ < U = const$



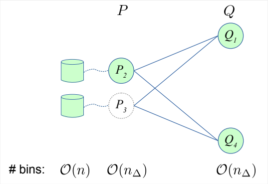
Efficient Computation of SND / EMD*

- Challenge: efficiently compute $\mathrm{EMD}^\star(P, Q, D)$ over sparse network
  - D (most-to-most shortest paths): $\mathcal{O}(n^2 \log n)$
  - $\mathrm{EMD}^\star$ (BP min-cost flow): $\mathcal{O}(n^3 \log n)$
- Assumption 1: number n_Δ of users who changed their opinions $\ll n$
- Assumption 2: $D_{ij} \in \mathbb{Z}^+ < U = const$
  - discard inactive bins



Efficient Computation of SND / EMD*

- Challenge: efficiently compute $\mathrm{EMD}^\star(P, Q, D)$ over sparse network
  - ▷ D (few-to-most shortest paths): $\mathcal{O}(n^2 \log n)$ $\mathcal{O}(n_\Delta n \log n)$
  - ▷ $\mathrm{EMD}^\star$ (BP min-cost flow): $\mathcal{O}(n^3 \log n)$
- Assumption 1: number n_Δ of users who changed their opinions $\ll n$
- Assumption 2: $D_{ij} \in \mathbb{Z}^+ < U = const$
  - ▷ discard inactive bins
  - ▷ discard bins having similar values ($\Leftarrow D$ is semimetric)



(unbalanced BP network)

Efficient Computation of SND / EMD*

- Challenge: efficiently compute $\mathrm{EMD}^\star(P, Q, D)$ over sparse network
  - ▷ D (few-to-most shortest paths): $\mathcal{O}(n^2 \log n)$ $\mathcal{O}(n_\Delta n \log \sqrt{U})$
  - ▷ $\mathrm{EMD}^\star$ (BP min-cost flow): $\mathcal{O}(n^3 \log n)$
- Assumption 1: number n_Δ of users who changed their opinions $\ll n$
- Assumption 2: $D_{ij} \in \mathbb{Z}^+ < U = const$
  - ▷ discard inactive bins
  - ▷ discard bins having similar values ($\Leftarrow D$ is semimetric)
  - ▷ use Dijkstra with radix + Fibonacci heaps[5] ($\Leftarrow$ Assumption 2)

[5] Ravindra K Ahuja et al. "Faster algorithms for the shortest path problem". In: *Journal of the ACM* 37.2 (1990), pp. 213–223.

Efficient Computation of SND / EMD*

- Challenge: efficiently compute $\mathrm{EMD}^\star(P, Q, D)$ over sparse network
  - D (few-to-most shortest paths): $\mathcal{O}(n^2 \log n)$ $\mathcal{O}(n_\Delta n \log \sqrt{U})$
  - $\mathrm{EMD}^\star$ (BP min-cost flow): $\cancel{\mathcal{O}(n^3 \log n)}$ $\mathcal{O}(n_\Delta m + n_\Delta^3 \log (n_\Delta n U))$
- Assumption 1: number n_Δ of users who changed their opinions $\ll n$
- Assumption 2: $D_{ij} \in \mathbb{Z}^+ < U = const$
  - discard inactive bins
  - discard bins having similar values ($\Leftarrow$ D is semimetric)
  - use Dijkstra with radix + Fibonacci heaps[5] ($\Leftarrow$ Assumption 2)
  - use modified Goldberg-Tarjan algorithm[6] ($\Leftarrow$ Assumptions 1, 2)

[5] Ahuja et al., "Faster algorithms for the shortest path problem".

[6] Ravindra K Ahuja et al. "Improved algorithms for bipartite network flow". In: *SIAM Journal on Computing* 23.5 (1994), pp. 906–933.

Efficient Computation of SND / EMD*

- Challenge: efficiently compute $\mathrm{EMD}^\star(P, Q, D)$ over sparse network
  - ▷ D (few-to-most shortest paths): $\mathcal{O}(n^2 \log n)$ $\mathcal{O}(n_\Delta n \log \sqrt{U})$
  - ▷ $\mathrm{EMD}^\star$ (BP min-cost flow): $\cancel{\mathcal{O}(n^3 \log n)}$ $\mathcal{O}(n_\Delta m + n_\Delta^3 \log{(n_\Delta n U)})$
- Assumption 1: number n_Δ of users who changed their opinions $\ll n$
- Assumption 2: $D_{ij} \in \mathbb{Z}^+ < U = const$
  - ▷ discard inactive bins
  - ▷ discard bins having similar values ($\Leftarrow D$ is semimetric)
  - ▷ use Dijkstra with radix + Fibonacci heaps[5] ($\Leftarrow$ Assumption 2)
  - ▷ use modified Goldberg-Tarjan algorithm[6] ($\Leftarrow$ Assumptions 1, 2)
- Achieved $T = \mathcal{O}(n_\Delta(n \log \sqrt{U} + n_\Delta^2 \log{(n_\Delta n U)}))$

---

[5] Ahuja et al., "Faster algorithms for the shortest path problem".
[6] Ahuja et al., "Improved algorithms for bipartite network flow".

Efficient Computation of SND / $\mathrm{EMD}^\star$

- Challenge: efficiently compute $\mathrm{EMD}^\star(P, Q, D)$ over sparse network
  - ▷ D (few-to-most shortest paths): $\mathcal{O}(n^2 \log n)$ $\mathcal{O}(n_\triangle n \log \sqrt{U})$
  - ▷ $\mathrm{EMD}^\star$ (BP min-cost flow): $\cancel{\mathcal{O}(n^3 \log n)}$ $\mathcal{O}(n_\triangle m + n_\triangle^3 \log (n_\triangle n U))$
- Assumption 1: number $n_\triangle$ of users who changed their opinions $\ll n$
- Assumption 2: $D_{ij} \in \mathbb{Z}^+ < U = const$
  - ▷ discard inactive bins
  - ▷ discard bins having similar values ($\Leftarrow$ D is semimetric)
  - ▷ use Dijkstra with radix + Fibonacci heaps[5] ($\Leftarrow$ Assumption 2)
  - ▷ use modified Goldberg-Tarjan algorithm[6] ($\Leftarrow$ Assumptions 1, 2)
- Achieved $T = \mathcal{O}(n_\triangle (n \log \sqrt{U} + n_\triangle^2 \log (n_\triangle n U)))$
- If $n_\triangle < const < \infty$, then $T = \mathcal{O}(n)$

[5] Ahuja et al., "Faster algorithms for the shortest path problem".
[6] Ahuja et al., "Improved algorithms for bipartite network flow".

Experimental Setting

- Synthetic data
  - scale-free network, $n = |V| = 10\text{k} \dots 200\text{k}$, $\gamma = -2.9 \dots - 2.1$
  - about equal number of initial adopters for $+$ and $-$
  - subsequent network states generated $\sim$ Independent Cascade
- Twitter data
  - crawled tweets mentioned "Obama" from May'08 to Aug'11
  - network of 10k politically-active users
  - each user has 130 neighbors, on average
  - user opinions are tracked over the entire period, quarter-wise
- Competing distance measures
  - hamming(P, Q)
  - quad-form$(P, Q, L) = \sqrt{(P - Q)L(P - Q)^T}$
  - walk-dist(P, Q): summarizes how different the network's users are from their respective neighbors

Application I: Anomaly Detection (Synthetic Data)

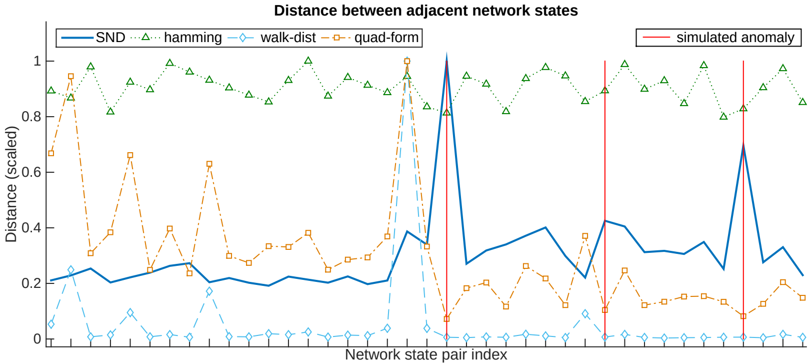


Figure: Anomaly detection on synthetic data. $|V| = 20$k, scale-free exponent $\gamma = -2.3$. A series of 40 network states is generated using $\mathbb{P}_{nbr} = 0.12$ and $\mathbb{P}_{ext} = 0.01$ for normal and $\mathbb{P}_{nbr} = 0.08$ and $\mathbb{P}_{ext} = 0.05$ for anomalous network states' generation, respectively. The three simulated anomalies are displayed as solid vertical lines.

- SND is good at detecting the anomalies not easily revealed just by looking at the rate of new user activation

Application I: Anomaly Detection (Synthetic Data)

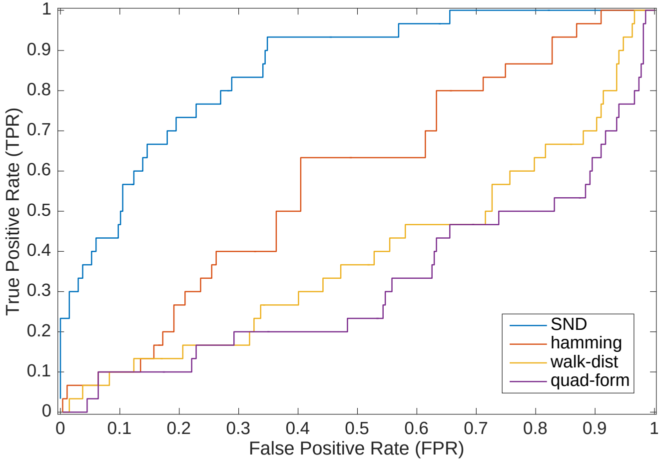


Figure: ROC curves comparing the quality of anomaly detection by different distance measures in a series of 300 network states over synthetic network with $|V| = 30k$ and scale-free exponent $\gamma = -2.3$. The network states are generated using $\mathbb{P}_{nbr} = 0.08$ and $\mathbb{P}_{ext} = 0.001$ for normal and $\mathbb{P}_{nbr} = 0.07$ and $\mathbb{P}_{ext} = 0.011$ for anomalous instances.

Application I: Anomaly Detection (Twitter Data)

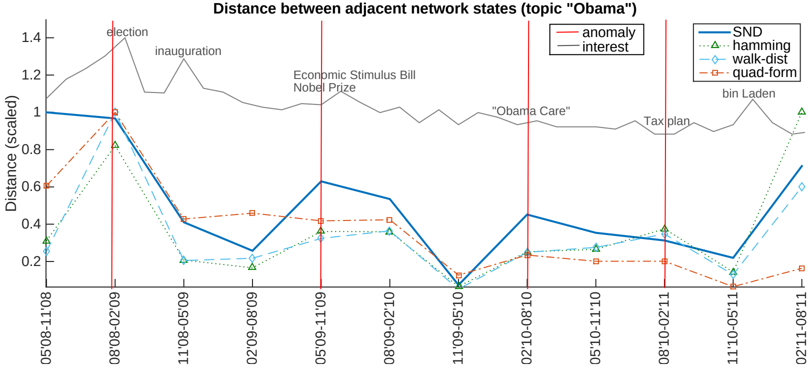


Figure: Anomaly detection on Twitter data (May'08-Aug'11). The distance series are accompanied by the curve showing Google Trends' scaled interest in topic "Obama". Network states detected to be anomalous by at least one distance measure are displayed as solid vertical lines.

- SND typically spikes and disagrees with other distance measures during "polarizing events" (e.g., "Obama Care")
- Events accompanied by drastic change in the rate of new user activation can be detected by any distance measure

Application II: User Opinion Prediction

- Given a series $G_0, G_1, \ldots, G_{t-1}, G_t$ of network states
- Goal: predict opinions of select users in G_t based on $G_{0\ldots t-1}$
- Approach
  - Compute distances (SND) between adjacent network states

$$d(G_0, G_1), \ldots, d(G_{t-2}, G_{t-1})$$

  - Extrapolate (LS) distance series to get expected $d_{exp} = d_{exp}(G_{t-1}, G_t)$
  - Assign opinions in G_t to minimize $|d(G_{t-1}, G_t) - d_{exp}|$
- Baselines
  - same approach with other distance measures
  - simulation until convergence (IC, LT) [Najar12]
  - (shallow) max-likelihood [Saito11]
  - based on community detection via label propagation [Conover11]

Application II: User Opinion Prediction

User Opinion Prediction Accuracy, %				
Method	Synthetic Data		Twitter Data	
	μ	σ	μ	σ
SND	**74.33**	**2.65**	**75.63**	**5.60**
hamming	68.44	12.34	68.13	5.80
quad-form	66.67	13.58	67.50	9.63
walk-dist	56.22	15.35	31.88	9.98
icc-simulation	**76.25**	**9.54**	59.38	4.17
ltc-simulation	67.50	11.65	58.75	5.18
icc-max-likelihood	67.41	7.03	57.50	8.02
ltc-max-likelihood	57.50	8.45	55.63	11.78
community-lp	65.25	9.43	56.87	8.43

Table: Means μ and standard deviations σ of user opinion prediction accuracies. Synthetic data generated using Independent Cascade.

Scalability of SND

- MATLAB/C++ implementation of SND publicly available (email us)
- Uses a simpler Dijkstra and an unmodified Goldberg-Tarjan
- Still scales well in practice

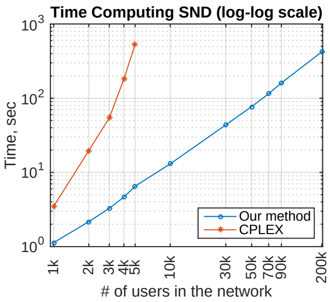


Figure: Time for computing SND when the number of users having different opinion is fixed at $n_\Delta = 1000$ and the total number of users n in the network grows up 200k.

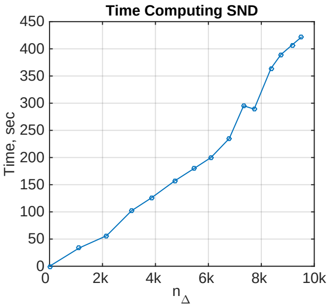


Figure: Time for computing SND using our method when the network size is fixed at $n = 20$k, and the number n_Δ of users having changed their opinions grows up to 10k.

Conclusion

- SND—first distance measure designed for the comparison of network states capturing dynamics of polar opinions.
- SND quantifies how likely it is that one state of a social network has evolved into another state under a given model of polar opinion propagation.
- It is computable in time linear in $|V|$, and, as such, applicable to real-world online social networks.
- In anomalous event detection, SND tends to detect well the events that have likely caused opinion polarization in the network. It is a good idea to use SND when simple summaries (e.g., number of new activations) are not informative enough.
- In user opinion prediction, SND performs reasonably well (75% accuracy), and outperforms baselines on real-world data.

Future Work

- Using SND in applications such as classification, clustering, and search.
- Extending SND to capture changes in *both* user opinions and network structure.

Implementation of SND

http://cs.ucsb.edu/~victor/pub/ucsb/dbl/snd/