Community-Enhanced De-anonymization of Online Social Networks

Shirin Nilizadeh, Apu Kapadia, Yong-Yeol Ahn
School of Informatics and Computing, Indiana University Bloomington

Online social networks have revolutionized the way our society communicates

But, social networking providers share (sell) 'anonymized' social network datasets

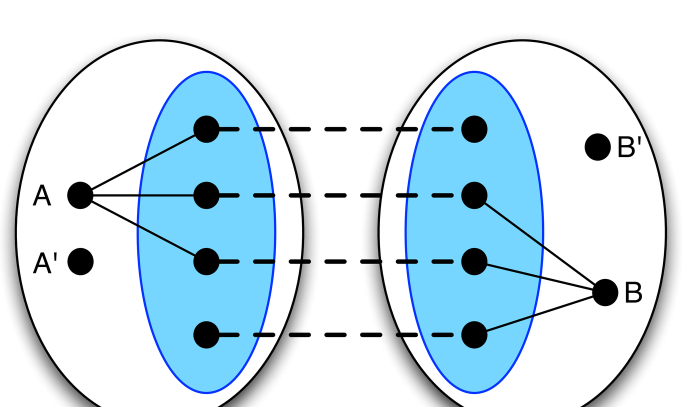
- OSN providers are treasure troves of information for marketers and researchers
- OSN provides release anonymized social networks to third-parties for various purposes including **targeted advertising**, **developing new applications, academic research, public competition**, etc.
- To protect the privacy of its users, social networking services attempt to 'anonymize' social network data, before sharing the datasets.
- For example, they provide the social-network structure but
  - Remove people's identities and
  - Add some 'noise' by modifying relationships and attributes to a certain extent.

Attack model

- We assume the recipient of this data, if malicious, may try to de-anonymize the social network
- We assume the adversary has access to two networks:
  - One of these networks is anonymized and contains sensitive private information associated with the (anonymized) nodes in the graph.
  - The other network is public (not anonymized) but does not contain any sensitive information
- **The goal of an attacker** is to re-identify anonymized users, and reveal the private information obtained from the anonymized network.

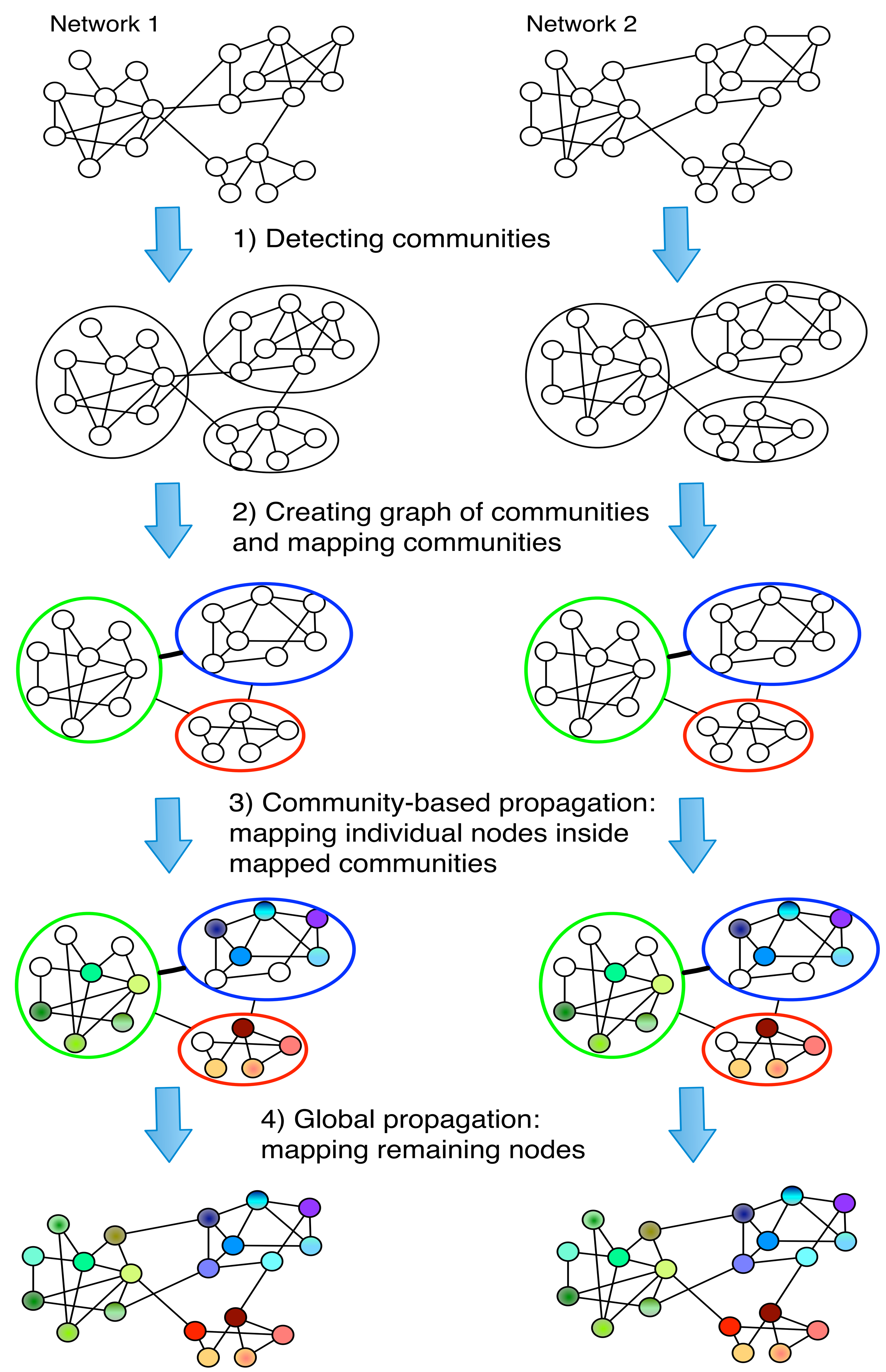
Re-identification algorithm by Narayanan and Shmatikov (NS)

1. **Seed identification** maps a small number of users (seeds) between two networks by searching for unique subgraphs.
2. **Propagation** expands the set of matched users by incrementally comparing and mapping the neighbors of the previously mapped seeds.



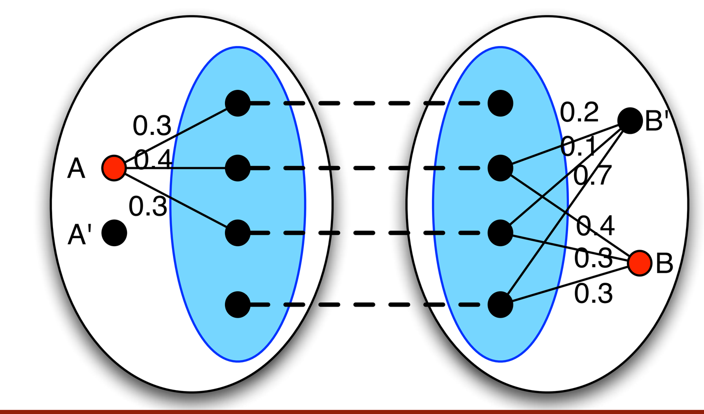
Community-enhanced De-anonymization

- We propose a 'mesoscopic' approach to improve the degree of de-anonymization.
- It divides the problem into smaller sub-problems that can be solved by leveraging existing network mapping methods recursively on multiple levels
  - First, it maps the community structure of two graphs by considering the community structure as a coarse-grained graph
  - It then applies the network mapping technique to the nodes inside each community (**Local propagation**) and finally to the entire graph (**Global propagation**)



Mapping communities by creating a network of communities

- We create a weighted undirected graph of communities, where,
  - each community is a node and
  - a weighted edge between two communities represents the number of connections between nodes in two communities



Seed enrichment

- Communities offer a much more narrow search space for seeds
- Two metrics
  - nodes' degrees (**d**), and,
  - the clustering coefficients (**cc**)

$$D_d(v_i, v_j) = \frac{|d(v_i) - d(v_j)|}{max(d(v_i), d(v_j))}$$

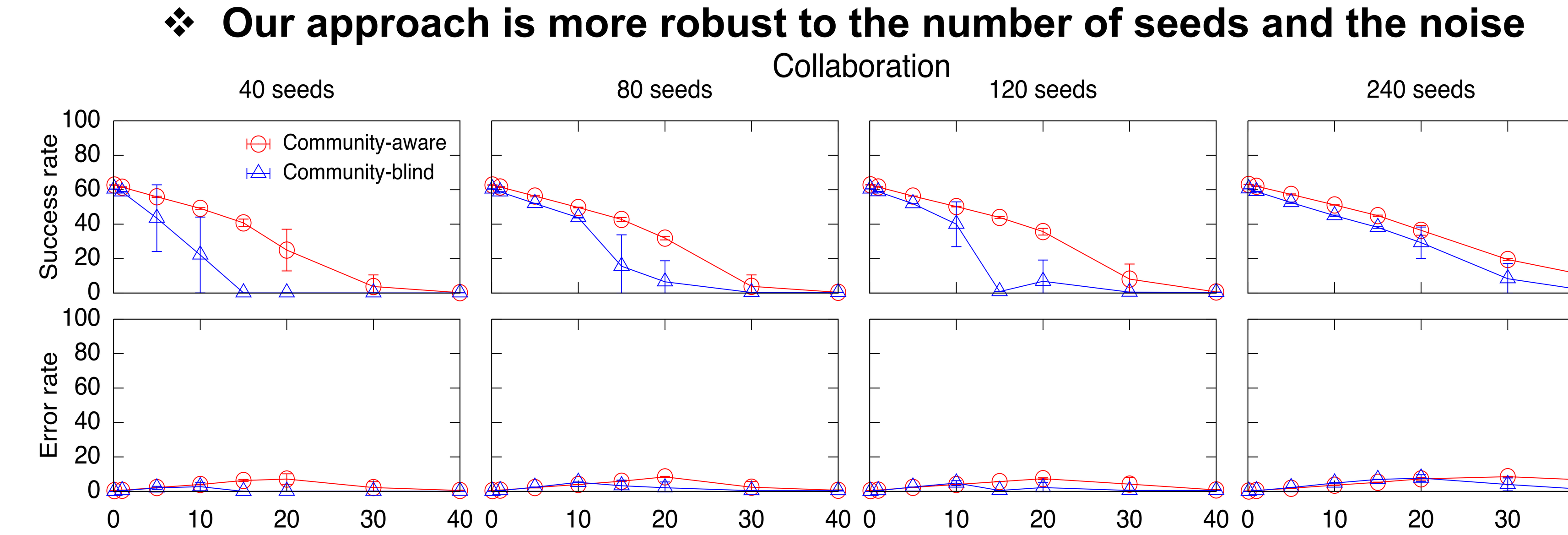
$$D_{cc}(v_i, v_j) = \frac{|cc(v_i) - cc(v_j)|}{max(cc(v_i), cc(v_j))}$$

Evaluation

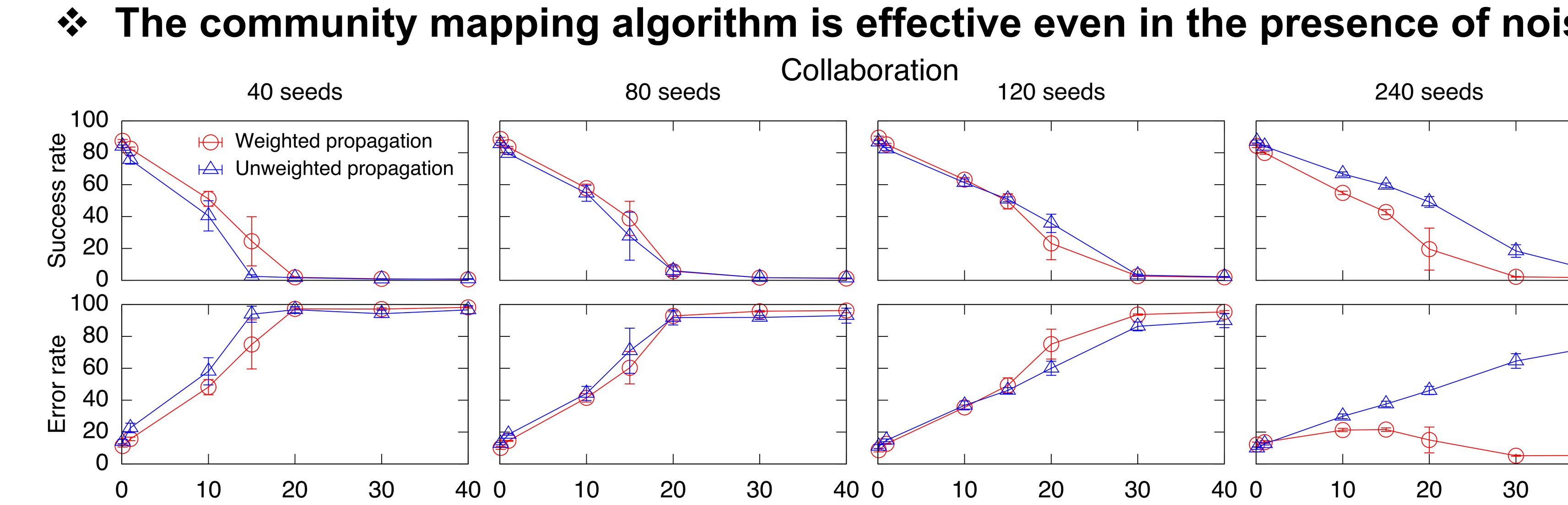
- We evaluate the performance of our approach by comparing it with the community-blind NS algorithm
- **Data Sets:**
  - Synthetic benchmark graphs (LFR-Benchmark generator)
  - Real-world graphs (collaboration network, and, Twitter mention Network)
- Generate noisy anonymized networks through edge rewiring
- **Performance metrics:**
  - **Success rate**: the percentage of correctly re-identified users
  - **Error rate** is the percentage of incorrectly mapped users
  - **Failure threshold** is the noise level that the algorithm starts to fail and provides no mapping
  - **Community mapping success rate** is the percentage of correctly mapped communities (based on Jaccard coefficient)
  - **Community mapping error rate**

Results

- **Our approach is more robust to the number of seeds and the noise**



- **The community mapping algorithm is effective even in the presence of noise**



INDIANA UNIVERSITY