Differentially Private Graph Learning via Sensitivity-Bounded Personalized PageRank

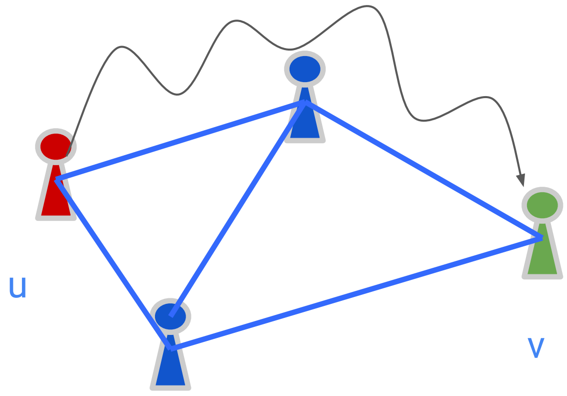
Alessandro Epasto Vahab Mirrokni Bryan Perozzi
Anton Tsitsulin *Peilin Zhong*

Google Research, New York

Personalized PageRank (PPR)

- Personalized PageRank (PPR): a standard tool in graph learning
    - In essence, **PPR** measures the **similarity of two nodes**, based on the network structure

- PPR of a source node u:
    - Random walk starting from u
    - Each step:
        - probability α returning back to u
        - probability 1-α proceeding to a random neighbor
    - PPR vector: stationary distribution of random walk
        - PPR value of v: probability staying in v



Applications

Applications in standard graph mining tasks:

- Link prediction; Recommender systems, Collaborative filtering
- Spam & Abuse detection; Anonymily detection
- Clustering

Recent ML applications:

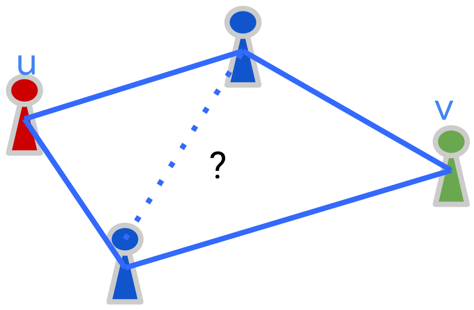
- Graph embeddings (InstantEmbedding)
- Efficiently running Graph-based Neural Network

Differentially Private PPR

- Extensive literature in approximating PPR in non-private settings

- No prior work on computing PPR in a differentially private (DP) way

- **Edge-ε-DP:** changing one edge in the graph has limited effect on the output distribution of the algorithm A
    - G and G' only differ in one edge
    - ∀ possible output set O, Pr[A(G)∈O] ≤ exp(ε) · Pr[A(G')∈O]



- This ensures that an attacker observing approximate PPR output will not learn about the existence of any specific edge in the graph

Theoretical Results

- First algorithms studying PPR under DP
  - The algorithms are always (joint) edge-ε-DP without any assumption of input graph

- **Edge-ε-DP PPR algorithm**
  - The output PPR has good approximation when the graph has uniformly large degree
    - The additive error is $O(1/D)$ where D is the minimum degree of the graph

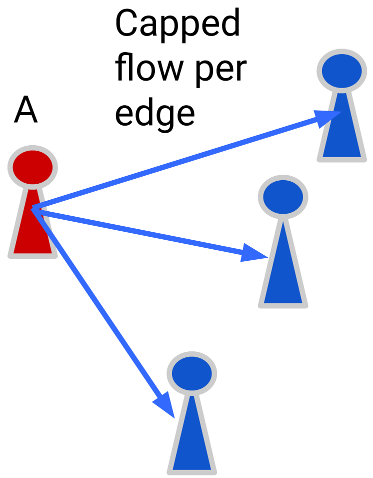
- **Joint edge-ε-DP PPR algorithm**
  - Joint edge-ε-DP: the different edge between neighboring graph cannot incident to source u
    - The private neighboring information of u is allowed to used to compute PPR of u
  - The additive error is $O(1/D^2)$

Main technical ideas

- We build upon the well-known **Push-Flow** algorithm for PPR to prove a **edge-sensitivity** bounded algorithm
  - Set a threshold that the total flow can be pushed along each edge
  - Laplace mechanism to get DP guarantees

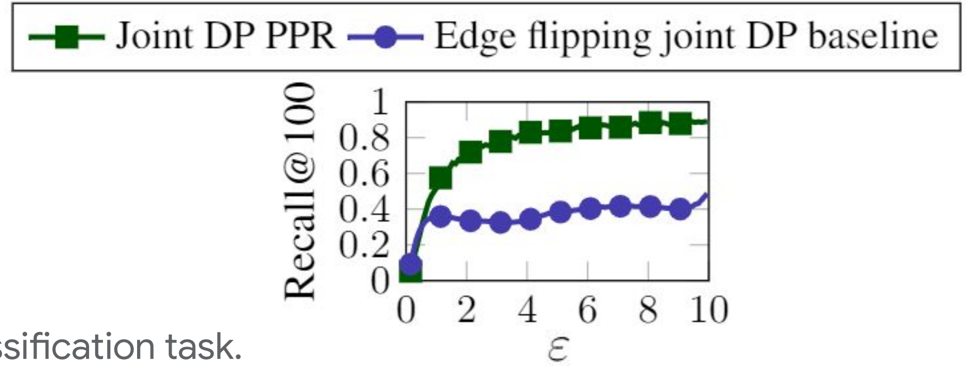
- We show that the dependency on the minimum degree is necessary
  - $\Omega(1/D)$ is necessary for edge-ε-DP
  - $\Omega(1/D^2)$ is necessary for joint edge-ε-DP

- This implies DP algorithms for downstream tasks using PPR: graph learning, embeddings etc



Experimental results

Example result for PPR ranking precision. Notice that the Joint DP algorithm has non-trivial recall even for small-ish epsilon values.



Example result for node classification task.