LOCALLY PRIVATE GRAPH NEURAL NETWORKS

Sina Sajadmanesh
Daniel Gatica-Perez

IDIAP RESEARCH INSTITUTE
SWISS FEDERAL INSTITUTE OF TECHNOLOGY (EPFL)

Twitter Machine Learning Seminar
Jan 7, 2021
Privacy-Preserving GNN learning with node-level privacy

**Setting:**
- The server has access to a graph
- Each node has a private feature vector
- Node features are inaccessible by the server

**Problem:**
- How to learn a GNN without letting the private features leave the nodes?
Can we use federated learning?

Can we use federated learning?
Can we use federated learning?
Can we use federated learning?

Can we use federated learning?

**INSIDE THE GNN**

**Input:** a representation vector $h_v$ for each node $v$  
(initially node features)

**Output:** a new representation vector $h'_v$ for each node $v$

---

Image Source: “A gentle introduction to graph neural networks” by Andreas Loukas.
**INSIDE THE GNN**

Input: a representation vector $h_v$ for each node $v$ (initially node features)

Output: a new representation vector $h'_v$ for each node $v$

$$h'_v = f(\{h_u : u \in \mathcal{N}(v)\})$$

$$= \text{UPDATE} \left( \text{AGGREGATE} \left( \{h_u : u \in \mathcal{N}(v)\} \right) \right)$$

- **UPDATE** is a neural network (e.g., MLP)
- **AGGREGATE** is a permutation invariant function, e.g., sum, mean, max, or:
  - **GCN:** $SUM \left( \frac{h_u}{\sqrt{\left| \mathcal{N}(u) \right| \cdot \left| \mathcal{N}(v) \right|}} : u \in \mathcal{N}(v) \right)$
  - **GraphSAGE:** $CONCAT \left( h_v, \text{MEAN} \left( \{h_u : u \in \mathcal{N}(v)\} \right) \right)$

*Image Source: “A gentle introduction to graph neural networks” by Andreas Loukas.*
**Input:** a representation vector $h_v$ for each node $v$

*initially node features*

**Output:** a new representation vector $h'_v$ for each node $v$

$$h'_v = f\left( \{ h_u : u \in \mathcal{N}(v) \} \right)$$

$$= \text{UPDATE} \left( \text{AGGREGATE} \left( \{ h_u : u \in \mathcal{N}(v) \} \right) \right)$$

- **UPDATE** is a neural network (e.g., MLP)
- **AGGREGATE** is a permutation invariant function, e.g., sum, mean, max, or:
  - **GCN:** $\text{SUM} \left( \frac{h_u}{\sqrt{|\mathcal{N}(u)| \cdot |\mathcal{N}(v)|}} : u \in \mathcal{N}(v) \right)$
  - **GraphSAGE:** $\text{CONCAT} \left( h_v, \text{MEAN} \left( \{ h_u : u \in \mathcal{N}(v) \} \right) \right)$

---

Image Source: “A gentle introduction to graph neural networks” by Andreas Loukas.
What's the problem with federated learning?

• **AGGREGATE** must be computed at node side
• Nodes require the private features of their neighbors
  • If sent in plain text → privacy violation!
  • If sent using SMC → massive communication!
Our Approach

Let’s keep the model on the server

- Private node features are only needed in the first layer of the GNN
- We only need to compute the first layer’s AGGREGATE function privately!

![Diagram](image-url)
Let's keep the model on the server

- Private node features are only needed in the first layer of the GNN.
- We only need to compute the first layer’s AGGREGATE function privately!
- But an exact computation of AGGREGATE is vulnerable to differencing attack!
Let’s keep the model on the server

- Private node features are only needed in the first layer of the GNN
- We only need to compute the first layer’s AGGREGATE function privately!
- But an exact computation of AGGREGATE is vulnerable to differencing attack!

Idea: privately estimate the AGGREGATE function using Local Differential Privacy!
Local Differential Privacy

- An **untrusted data aggregator** wishes to compute an aggregate function over a **private dataset**
- Data holder $i$ perturbs his data $x_i$ using a **randomized mechanism** $\mathcal{M}$, returning $x'_i = \mathcal{M}(x_i)$ to the aggregator
- The aggregator computes the target statistic using an **estimator function**
Local Differential Privacy

- An **untrusted data aggregator** wishes to compute an aggregate function over a **private dataset**
- Data holder $i$ perturbs his data $x_i$ using a **randomized mechanism** $\mathcal{M}$, returning $x'_i = \mathcal{M}(x_i)$ to the aggregator
- The aggregator computes the target statistic using an **estimator function**

**Definition**

A randomized mechanism $\mathcal{M}$ satisfies $\epsilon$-LDP if for all pairs of private data $x_1$ and $x_2$, and for all outputs $x'$ of $\mathcal{M}$, we have:

$$\Pr[\mathcal{M}(x_1) = x'] \leq e^\epsilon \Pr[\mathcal{M}(x_2) = x']$$
Private neighborhood aggregation with LDP

- Node features are perturbed by **injecting noise**
  - The LDP mechanism should be **unbiased**, i.e., $\mathbb{E}[\mathcal{M}(x)] = x$
  - The neighborhood aggregation will **cancel out** the injected noise
- **AGGREGATE** should be a **weighted summation**, as in GCN, GIN, ...
Locally Private GNN (LPGNN)

Node-Side:
1. Perturb the private feature vector $x_i$ using the LDP mechanism $x_i' = M(x_i)$ and send $x_i'$ to the server.

Server-Side:
1. Estimate the first layer’s AGGREGATE function for every node using the perturbed feature vectors.
2. Proceed with forward and backward propagation as usual.
3. Return to step 2 if stopping criteria has not met.
Challenge #1: High-dimensional features

- The total privacy budget of a node scales with the number of features
  - Give every single feature a small privacy budget → **Too much privacy leakage!**
  - Keep the total privacy budget of a node small → **Too much noise!**
Multi-bit mechanism: multidimensional feature perturbation

- Randomly sample $m/d$ features without replacement
- Perturb each sampled feature with $\epsilon/m$ privacy budget using 1-bit mechanism
- Map the output of the 1-bit mechanism to either -1 or 1
- For the rest of the features (not sampled) return 0

Theorem 3.1

The multi-bit mechanism satisfies $\epsilon$-LDP for each node.

Proposition 3.5

The optimal value of the sampling parameter $m$ in the multi-bit mechanism is:

$$m^* = \max(1, \min(d, \left\lfloor \frac{\epsilon}{2.18} \right\rfloor))$$
Challenge #2: Small-size neighborhood

- Lots of the nodes have **too few neighbors** (Power-Law degree distribution)
- The neighborhood aggregator **cannot cancel out the noise** if the neighborhood size is small
**KProp convolution layer:** neighborhood expansion method

- Expands the neighborhood to the nodes that are up to K-hops away
- Applies $K$ consecutive **AGGREGATE** function
- Applies the **UPDATE** function after the $K$-th **AGGREGATE**
- Trade-off between the aggregation estimation error and over-smoothing
Learning Task

• Node Classification

Comparison methods

• **GCN+RAW**: A standard two-layer GCN trained on raw features (non-private)
• **LPGNN**: A two-layer GNN (KProp as the first, GCN as the second layer) trained on perturbed features using the multi-bit mechanism (locally-private)
• **GCN+RND**: Similar to GCN+RAW, but trained on random features (fully-private)
• **GCN+OHD**: Similar to GCN+RAW, but trained on “one-hot degree” features (fully-private)
### Experiments: Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Classes</th>
<th>#Features</th>
<th>Avg. Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORA</td>
<td>7</td>
<td>1,433</td>
<td>3.90</td>
</tr>
<tr>
<td>CITESEER</td>
<td>6</td>
<td>3,703</td>
<td>2.74</td>
</tr>
<tr>
<td>PUBMED</td>
<td>3</td>
<td>500</td>
<td>4.50</td>
</tr>
<tr>
<td>FACEBOOK</td>
<td>4</td>
<td>4,714</td>
<td>15.21</td>
</tr>
<tr>
<td>GITHUB</td>
<td>2</td>
<td>4,005</td>
<td>15.33</td>
</tr>
<tr>
<td>LASTFM</td>
<td>18</td>
<td>7,842</td>
<td>7.29</td>
</tr>
</tbody>
</table>
## Results: Analysis of Accuracy-Privacy Trade-Off

### Micro-F1 score of different methods for node classification

<table>
<thead>
<tr>
<th>Dataset</th>
<th>GCN +Raw</th>
<th>LPGNN $\epsilon = 0.1$</th>
<th>LPGNN $\epsilon = 0.5$</th>
<th>LPGNN $\epsilon = 1.0$</th>
<th>LPGNN $\epsilon = 2.0$</th>
<th>GCN +RND</th>
<th>GCN +OHD</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORA</td>
<td>85.0 ± 0.5</td>
<td>84.6 ± 0.5</td>
<td>84.6 ± 0.6</td>
<td>84.6 ± 0.6</td>
<td>84.6 ± 0.6</td>
<td>78.1 ± 1.3</td>
<td>58.4 ± 0.7</td>
</tr>
<tr>
<td>CITESEER</td>
<td>73.7 ± 0.5</td>
<td>68.6 ± 0.8</td>
<td>68.4 ± 0.7</td>
<td>68.6 ± 0.9</td>
<td>68.6 ± 0.8</td>
<td>58.3 ± 4.1</td>
<td>38.5 ± 0.9</td>
</tr>
<tr>
<td>PUBMED</td>
<td>87.0 ± 0.2</td>
<td>82.1 ± 0.2</td>
<td>82.2 ± 0.3</td>
<td>82.2 ± 0.3</td>
<td>82.2 ± 0.3</td>
<td>56.5 ± 2.2</td>
<td>62.4 ± 0.9</td>
</tr>
<tr>
<td>FACEBOOK</td>
<td>94.8 ± 0.1</td>
<td>94.0 ± 0.1</td>
<td>94.0 ± 0.2</td>
<td>94.0 ± 0.2</td>
<td>94.0 ± 0.2</td>
<td>40.6 ± 1.2</td>
<td>79.2 ± 0.3</td>
</tr>
<tr>
<td>GITHUB</td>
<td>86.7 ± 0.2</td>
<td>85.9 ± 0.1</td>
<td>85.9 ± 0.2</td>
<td>85.9 ± 0.2</td>
<td>85.9 ± 0.1</td>
<td>74.4 ± 0.1</td>
<td>84.0 ± 0.1</td>
</tr>
<tr>
<td>LASTFM</td>
<td>87.7 ± 0.4</td>
<td>86.1 ± 0.3</td>
<td>86.1 ± 0.3</td>
<td>86.1 ± 0.2</td>
<td>86.1 ± 0.3</td>
<td>25.2 ± 7.1</td>
<td>70.6 ± 0.5</td>
</tr>
</tbody>
</table>
RESULTS: EFFECT OF THE MULTI-BIT MECHANISM

Average L1 distance between the true and estimated neighborhood aggregation obtained by the multi-bit (MBM), 1-bit (1BM), and the Analytic Gaussian (AGM) mechanisms
Effect of the KProp step parameter ($K$) on the performance of LPGNN ($\epsilon = 1$)

**PUBMED**

<table>
<thead>
<tr>
<th>Step Parameter ($K$)</th>
<th>GCN+RAW</th>
<th>LPGNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^0$</td>
<td>84</td>
<td>77</td>
</tr>
<tr>
<td>$2^1$</td>
<td>83</td>
<td>78</td>
</tr>
<tr>
<td>$2^2$</td>
<td>82</td>
<td>80</td>
</tr>
<tr>
<td>$2^3$</td>
<td>81</td>
<td>81</td>
</tr>
<tr>
<td>$2^4$</td>
<td>82</td>
<td>82</td>
</tr>
<tr>
<td>$2^5$</td>
<td>83</td>
<td>83</td>
</tr>
</tbody>
</table>

**FACEBOOK**

<table>
<thead>
<tr>
<th>Step Parameter ($K$)</th>
<th>GCN+RAW</th>
<th>LPGNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^0$</td>
<td>94</td>
<td>91</td>
</tr>
<tr>
<td>$2^1$</td>
<td>93</td>
<td>92</td>
</tr>
<tr>
<td>$2^2$</td>
<td>92</td>
<td>93</td>
</tr>
<tr>
<td>$2^3$</td>
<td>91</td>
<td>94</td>
</tr>
<tr>
<td>$2^4$</td>
<td>90</td>
<td>93</td>
</tr>
<tr>
<td>$2^5$</td>
<td>89</td>
<td>92</td>
</tr>
</tbody>
</table>
Effect of the label rate on the performance of LPGNN ($\epsilon = 1$)

**Results: Effect of the Label Rate**

Effect of the label rate on the performance of LPGNN ($\epsilon = 1$)

![Graph showing Micro-F1 % vs. Label Rate for PubMed and Facebook datasets with different K values (K = 1, K = 4, K = 8)]
Summary

- Proposed a privacy-preserving GNN based on local differential privacy
- Demonstrated promising results in terms of accuracy-privacy trade-off
- Works better on graphs with higher average degree

Future Work

- Protect privacy of graph topology
- Relationship to adversarial robustness
- Expressive power of private graph networks
THANK YOU!

Questions? @sajadmanesh
sajadmanesh@idiap.ch