Privacy-Preserving Graph Convolutional Networks for Text Classification

Timour Igamberdiev and Ivan Habernal
Trustworthy Human Language Technologies (TrustHLT), Ubiquitous Knowledge Processing Lab (UKP)
Department of Computer Science, Technische Universität Darmstadt
`https://www.trusthlt.org`, `https://www.ukp.tu-darmstadt.de`

Introduction

- Many text classification datasets naturally occur as graphs
  - Nodes: Text documents
  - Edges: Task specific (e.g. documents citing each other)
- Graph convolutional networks (GCNs) [1]: Powerful architecture for such tasks
- However, machine learning models do not protect privacy, possible for adversary to reveal sensitive information from training data (e.g. membership inference [2])
- Possible solution: Differentially private [3] machine learning with DP-SGD [4]
- Issue: Algorithm expects data examples to form batches and lots, not possible for large one-graph datasets
- Our solution: Graph splitting approach for adapting GCNs in the DP setting

Background on Differential Privacy

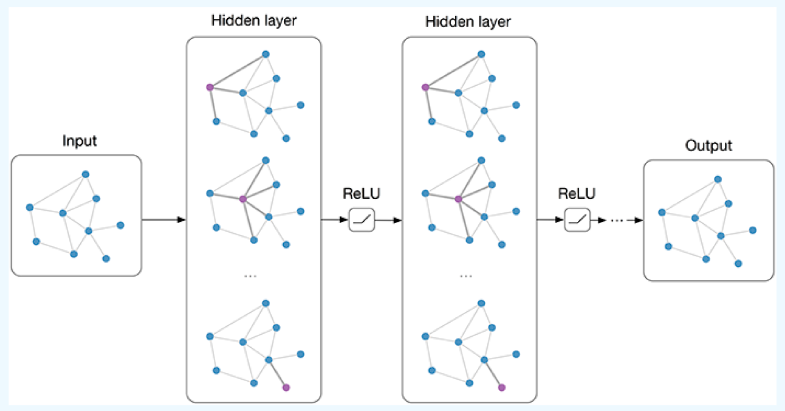
- Data inputs/outputs are perturbed to form a mathematically rigorous privacy guarantee
- The output of an algorithm or query is indistinguishable when adding or removing an individual from the dataset

$$\boxed{Pr[A(D) \in S] \leq \exp(\varepsilon) Pr[A(D') \in S] + \delta}$$

for two neighboring datasets D and D', a randomized algorithm A, and set of outputs S

Methods

- Our underlying architecture: Vanilla GCN [1]
- DP-SGD: Noise added to clipped gradient of a network during training
- DP-Adam: Extension of DP-SGD for Adam optimizer [5]
- Need a lot of noise to preserve privacy of graph datasets, without a way to split into batches and lots required for DP-SGD



$$\tilde{\mathbf{g}}_t = \frac{1}{L}\left(\sum_{i \in L} \frac{\mathbf{g}_t(x_i)}{\max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)} + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$$

Our Solution

Random graph partitioning:

- Create a random index tensor for all nodes in the training set
- Split into s groups, with s being the desired number of subgraphs
- Use the resulting indexes to mask the original graph during training

Experimental Setup

Dataset	Classes	Test size	Training size
CiteSeer	6	1,000	1,827
Cora	7	1,000	1,208
PubMed	3	1,000	18,217
Pokec	2	2,000	16,000
Reddit	41	5,643	15,252

- Languages: English and Slovak
- Features: BoWs (CiteSeer, Cora, PubMed), GloVe (Reddit), BERT (Pokec)
- Experiments: Graph partitioning, varying size of training data, DP vs. non-DP

Results: Without Graph Cuts

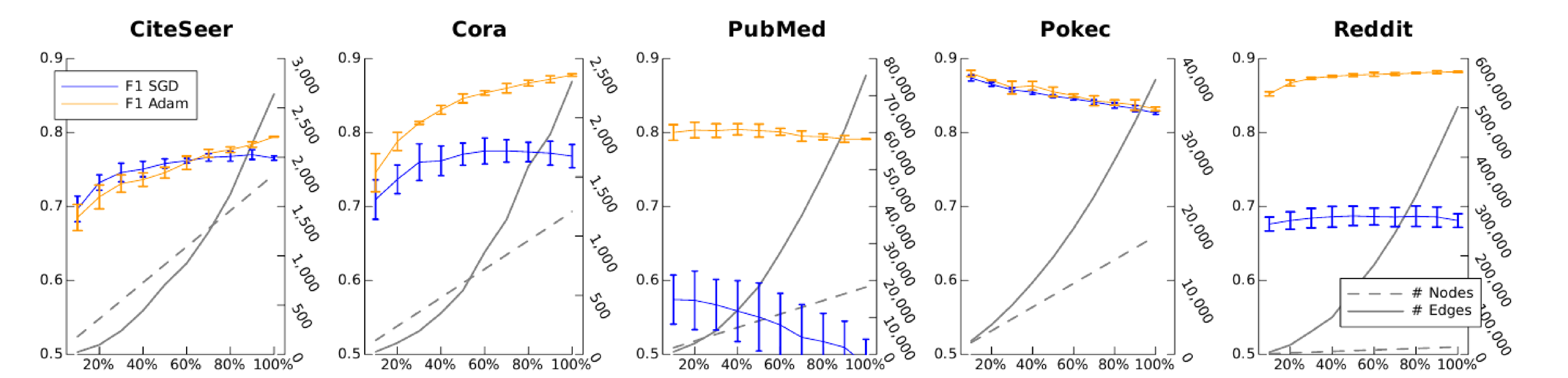


Figure 1: Exp. A: F_1 wrt. training data size (in %), without DP

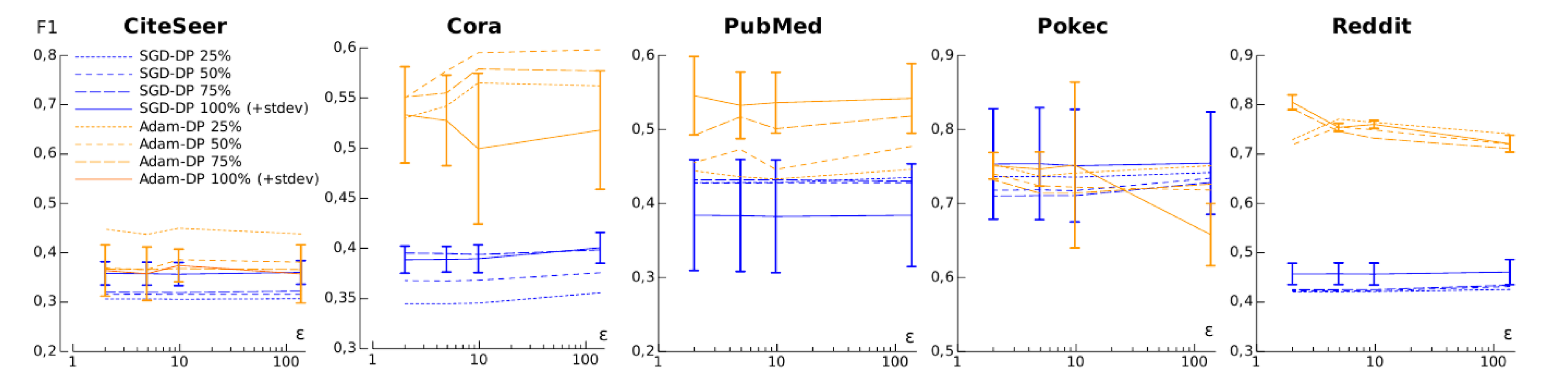


Figure 2: Exp. B: F_1 wrt. varying training data size (in %) wrt. privacy budget ε, with DP

Results: Graph Cuts

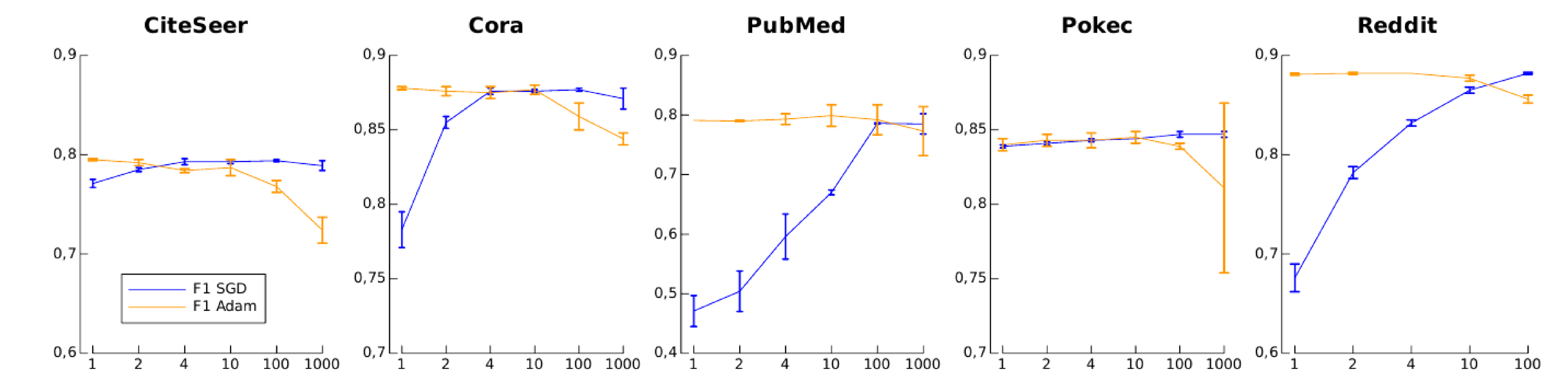


Figure 3: Exp. C, no DP: F_1 wrt. number of subgraphs

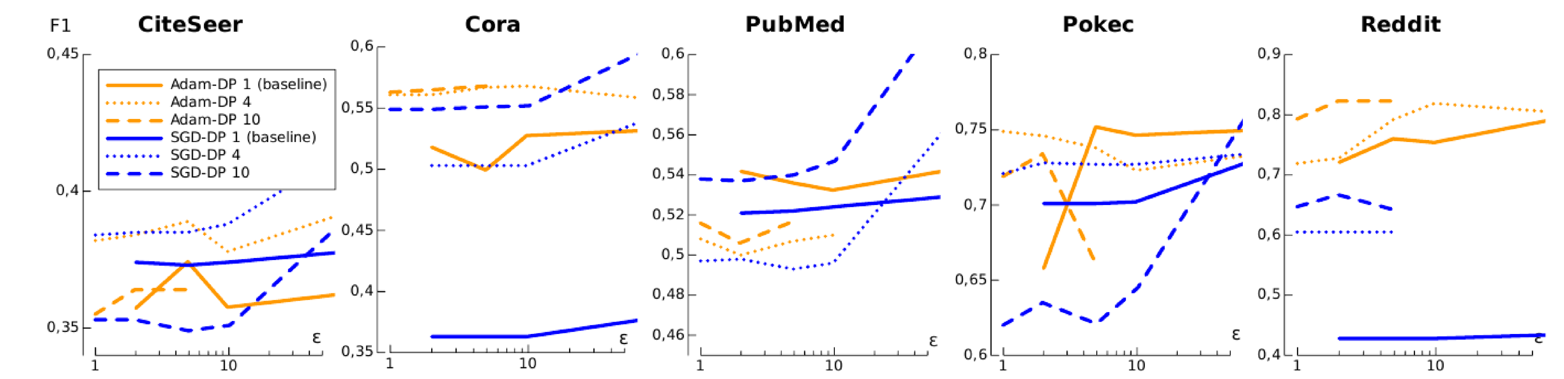


Figure 3: Exp. C, with DP: F_1 with varying number of subgraphs wrt. privacy budget ε

Results: Summary

	Non-DP				DP		DP split	
	Maj.	SGD	Adam	ε	SGD	Adam	SGD	Adam
CiteSeer				1	-	-	0.35	0.36
	0.18	0.77	0.79	2	**0.36**	**0.36**	0.35	**0.36**
Cora				1	-	-	0.55	0.56
	0.32	0.77	0.88	2	0.39	0.52	0.55	**0.57**
PubMed				1	-	-	0.54	0.52
	0.40	0.49	0.79	2	0.38	**0.54**	**0.54**	0.51
Pokec				1	-	-	0.62	0.72
	0.50	0.83	0.83	2	**0.75**	0.66	0.64	0.73
Reddit				1	-	-	0.65	0.79
	0.15	0.68	0.88	2	0.46	0.72	0.67	**0.82**

- Graph partitioning improves both performance and allows for a stronger privacy guarantee of $\varepsilon = 1$
- Increasing training data does not necessarily mitigate negative performance of DP
- More complex representations better for DP setting, with a smaller drop from non-DP results

Contact

Email: `timour.igamberdiev@tu-darmstadt.de`