TNE: A Latent Model for Representation Learning on Networks

Abdulkadir Çelikkanat and Fragkiskos D. Malliaros

Center for Visual Computing
CentraleSupélec, University of Paris-Saclay and Inria Saclay

Introduction

- *Network representation learning* (NRL) aims to encode a given network structure into low-dimensional vectors
- Applications in network analysis: visualization, classification, community detection and link prediction
- Proposed method:
  - TNE – Topical Node Embeddings
  - Enriched feature vectors using node and community information

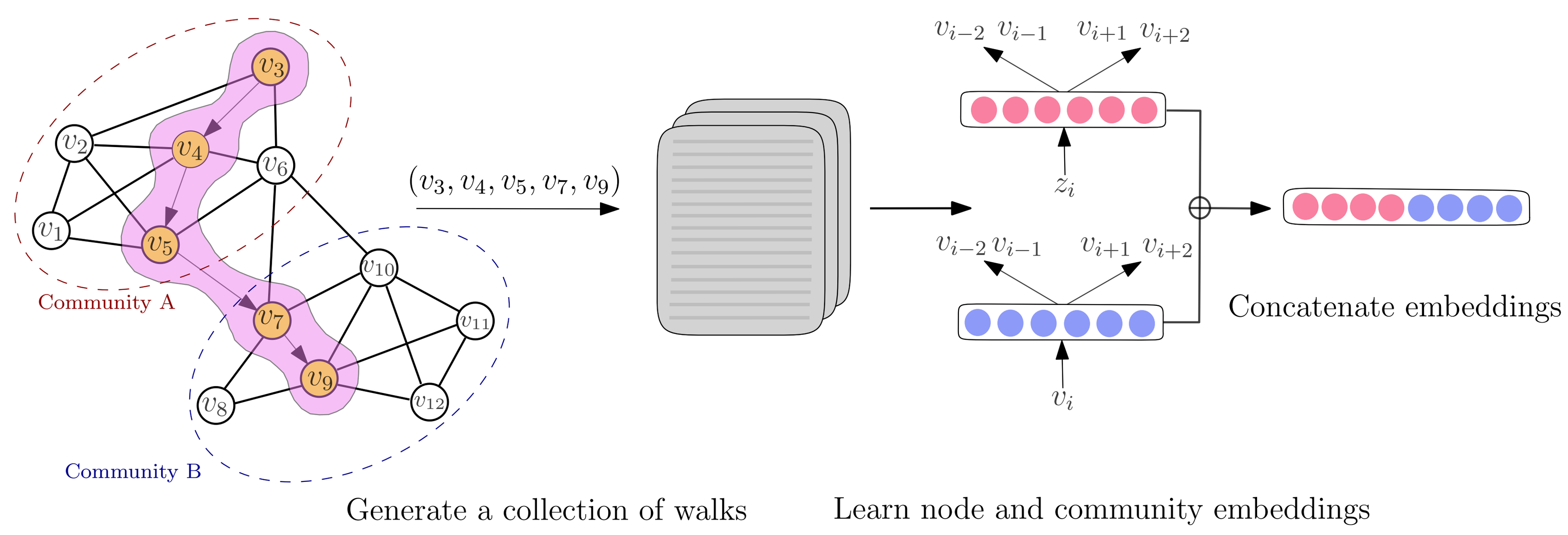

Figure: Schematic representation of the TNE model

Problem Formulation

Objective

For a given graph $G = (\mathcal{V}, \mathcal{E})$, the goal is to find a mapping function

$$\Phi : \mathcal{V} \rightarrow \mathbb{R}^d,$$

where $\Phi(v)$ indicates the representation of the vertex v in $\mathbb{R}^d$

- The objective function of random walk-based methods is:

$$\max_{\Phi,\widetilde{\Phi}} \sum_{v} \sum_{u \in N_\gamma(v)} \log \Pr(\Phi(u)|\widetilde{\Phi}(v)),$$

where $N_\gamma(v)$ is the set of reachable nodes starting from node $v \in \mathcal{V}$ in at most γ steps

- Approximation of the objective function:

$$\max_{\Phi,\widetilde{\Phi}} \sum_{w \in \mathcal{W}} \sum_{v_i \in w} \sum_{-\gamma \leq j \leq \gamma} \log \Pr(\Phi(v_{i+j})|\widetilde{\Phi}(v_i))$$

Random Walks and Communities

Can we take advantage of the clustering structure of the graph?

Random walk-based graph topic models

- *tne-LDA*
  - Each random walk can be represented as random mixtures over latent communities
  - Each community can be characterized by a distribution over nodes
- *tne-HMM*
  - The hidden state of the current node can also be utilized towards determining the next node to visit

Network structure-based modeling

- *tne-Louvain*
  - Community detection based on modularity opt.
- *tne-BigClam*
  - Overlapping community detection algorithm

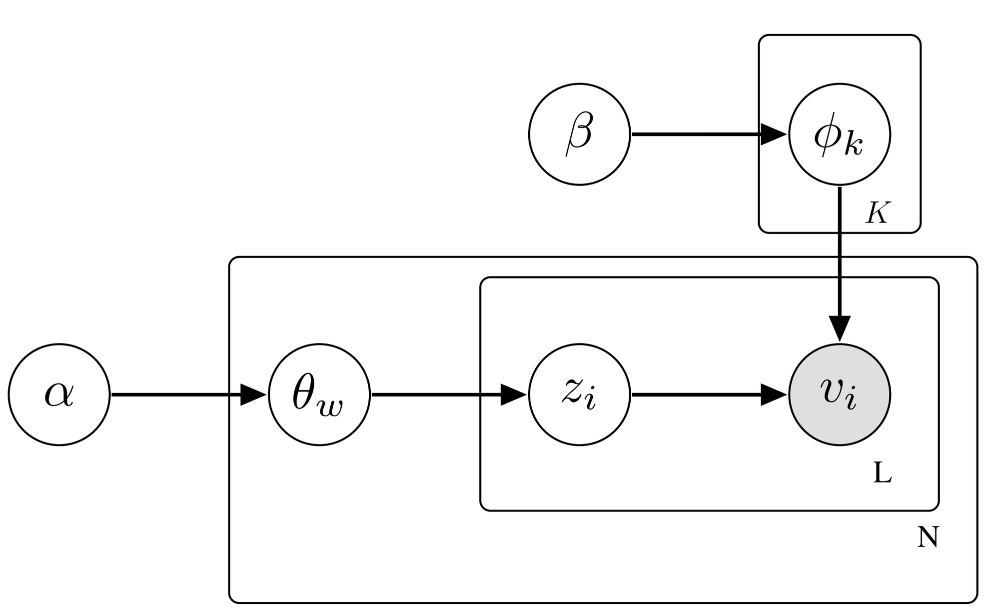

Figure: Graphical representation of the LDA model

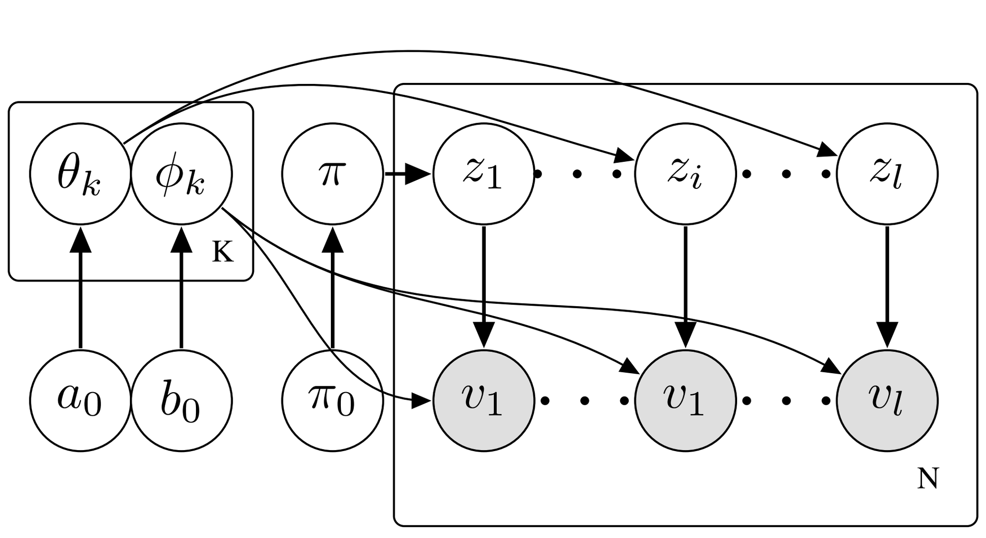

Figure: Graphical representation of non-parametric Hidden Markov Model (HMM) model

Topical Node Embeddings (TNE)

- Let t_v^w be a community/topic assignment of a node v in the walk $w \in \mathcal{W}$
- The objective function to learn *topic embeddings* is:

$$\max_{\Psi,\widetilde{\Psi}} \sum_{w \in \mathcal{W}} \sum_{v_i \in w} \sum_{-\gamma \leq j \leq \gamma} \log \Pr(\Psi(v_{i+j})|\widetilde{\Psi}(t_i^w))$$

- The final embedding vector is obtained by combining node and community embeddings

$$\Phi(v) \oplus \widetilde{\Psi}(k^*) \quad \text{where } k^* = \arg\max_k \Pr(\Phi(u)|\widetilde{\Psi}(k))$$

Experimental Results

	CiteSeer	Cora	PPI	Gnutella	FB	arXiv
# Vertices	3,312	2,708	3,890	8,114	4,039	5,242
# Edges	4,660	5,278	38,739	26,013	88,234	14,496
# Clusters	6	7	50	-	-	-

Table: Networks used in the experiments

- **Multi-label Node Classification**

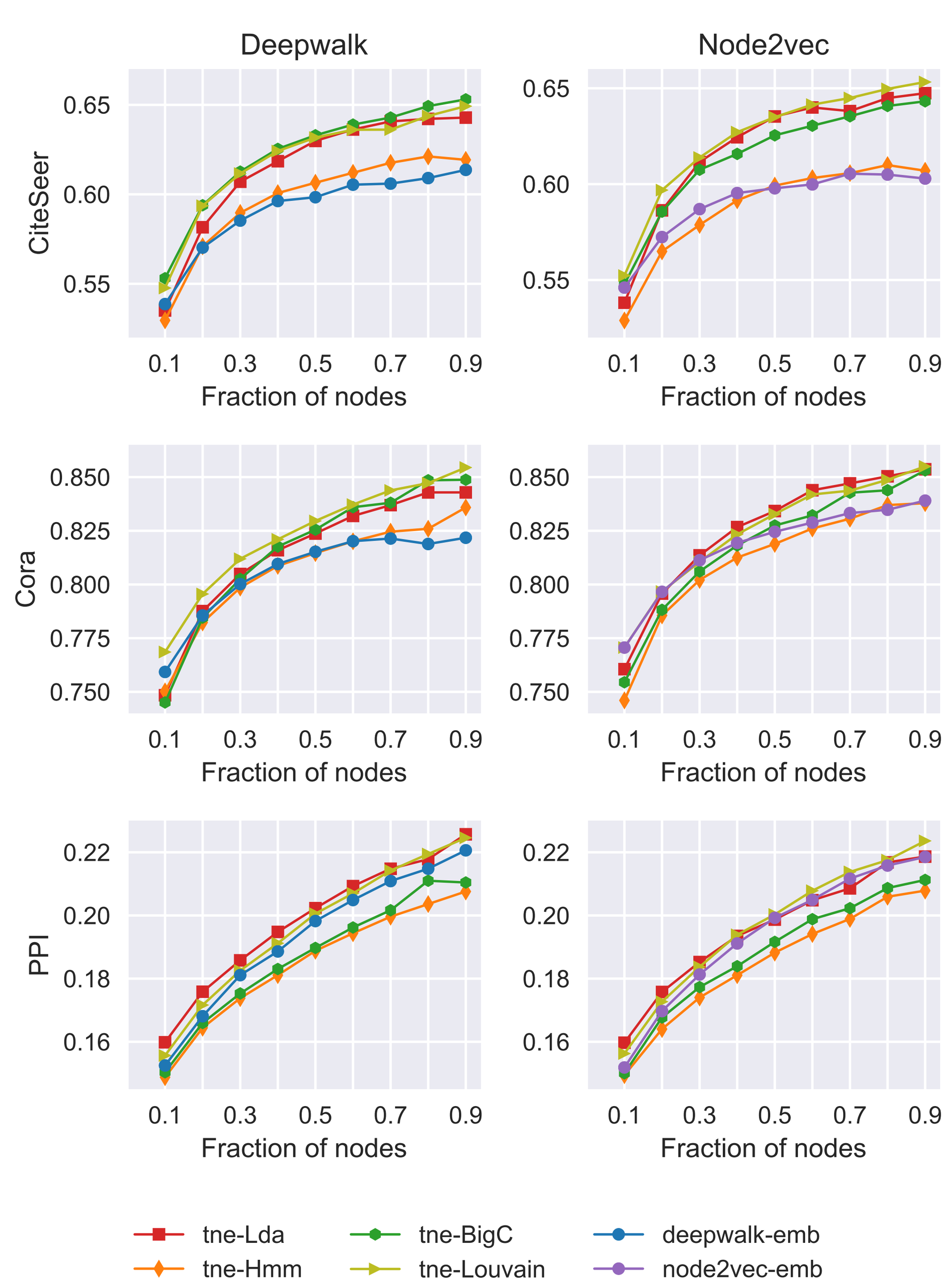

Figure: Micro-F_1 scores for multi-label node classification over three different networks

		Baseline	*tne-Lda*	*tne-Hmm*	*tne-BigC*	*tne-Louvain*
Citeseer	*Deepwalk*	0.554	0.590	0.565	0.591	0.589
	Gain/Loss (%)		**6.58**	**2.02**	**6.69**	**6.45**
	Node2vec	0.551	0.591	0.556	0.586	0.593
	Gain/Loss (%)		**7.32**	**0.84**	**6.31**	**7.58**
Cora	*Deepwalk*	0.808	0.816	0.807	0.814	0.819
	Gain/Loss (%)		**1.04**	**-0.03**	**0.81**	**1.42**
	Node2vec	0.814	0.822	0.807	0.817	0.823
	Gain/Loss (%)		**0.96**	**-0.93**	**0.28**	**1.10**
PPI	*Deepwalk*	0.174	0.179	0.165	0.168	0.175
	Gain/Loss (%)		**2.83**	**-5.01**	**-3.14**	**0.80**
	Node2vec	0.174	0.175	0.164	0.169	0.173
	Gain/Loss (%)		**0.47**	**-5.68**	**-2.90**	**-0.47**

Table: Macro-F_1 scores for node classification, where 50% of nodes are used for training

- **The effect of the number of topics/communities**

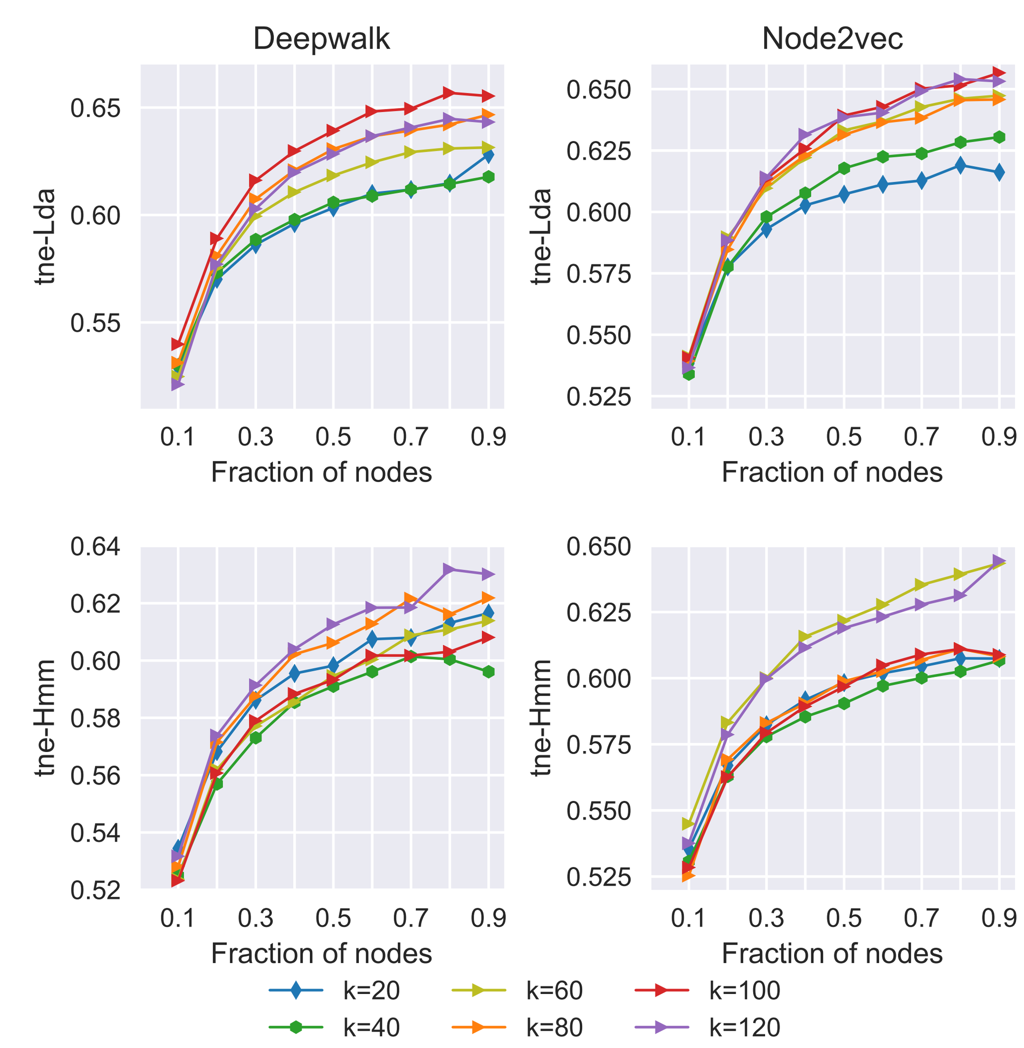

Figure: Varying number of topics/communities over CiteSeer

- **Link Prediction**

		(a) dw	(a) n2v	(b) dw	(b) n2v	(c) dw	(c) n2v
Gnutella	*Baseline*	0.705	0.714	0.582	0.619	0.579	0.617
	tne-Lda	0.704	0.708	0.585	0.622	0.582	0.620
	tne-Hmm	**0.712**	**0.726**	0.582	0.617	0.573	0.613
	tne-BigC	0.704	0.722	0.586	**0.627**	0.580	**0.625**
	tne-Louvain	0.699	0.707	**0.587**	0.618	**0.582**	0.615
Facebook	*Baseline*	0.753	0.750	0.983	0.983	0.984	0.983
	tne-Lda	0.777	0.774	0.985	0.986	0.986	0.986
	tne-Hmm	**0.778**	**0.778**	0.986	0.986	0.986	0.986
	tne-BigC	0.771	0.773	**0.986**	**0.986**	**0.987**	**0.986**
	tne-Louvain	0.759	0.761	0.984	0.984	0.984	0.984
ArXiv gr-qc	*Baseline*	0.725	0.725	0.924	0.930	0.925	0.931
	tne-Lda	0.723	0.724	**0.933**	**0.931**	**0.933**	0.932
	tne-Hmm	0.722	0.729	0.920	0.930	0.921	0.932
	tne-BigC	**0.727**	0.731	0.923	0.928	0.922	0.929
	tne-Louvain	0.732	**0.737**	0.927	0.934	0.926	**0.934**

Table: AUC scores for the link prediction task with operators: (a) Average, (b) Weighted-L1, and (c) Weighted-L2. (dw: Deepwalk, n2v: Node2vec)

Operator	Definition
Average	$0.5 \cdot (v + u)$
Weighted-L1	$\lvert v - u \rvert_1$
Weighted-L2	$\lvert v - u \rvert_2$

Table: Operators for learning edge features

References

[1] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Topical word embeddings. In *AAAI*, 2015.

[2] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD*, 2014.

[3] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *KDD*, 2016.

[4] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech.*, 2008.

[5] Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: A nonnegative matrix factorization approach. In *WSDM*, 2013.

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.