Kernel Node Embeddings

Abdulkadir Çelikkanat and Fragkiskos D. Malliaros

CentraleSupélec, Inria, Université Paris-Saclay

Introduction

- *Network representation learning* (NRL) aims to encode the structure of a network into low-dimensional vectors
- Applications in network analysis: visualization, classification, community detection and link prediction
- In this work:
  - We propose a novel approach for learning node embeddings by incorporating kernel functions with models relying on weighted matrix factorization
  - We perform extensive performance evaluation of the proposed method in two downstream tasks

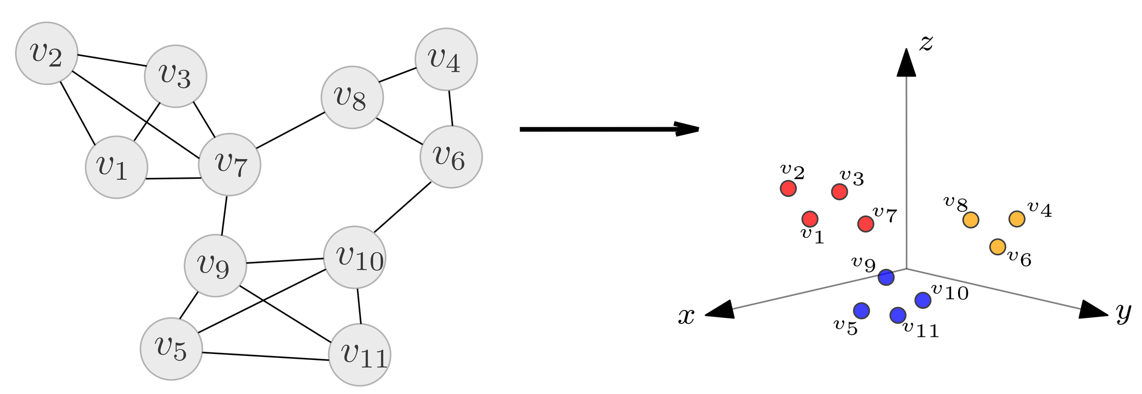

Figure: Schematic representation of node embeddings

Proposed Approach

- We define the general objective function of our problem as a weighted matrix factorization:

$$\arg\min_{\mathbf{A},\mathbf{B}} \frac{1}{2}\left\|\mathbf{W}\odot(\mathbf{M}-\mathbf{A}\mathbf{B}^\top)\right\|_F^2 \quad (1)$$

- By setting each term $\mathbf{W}_{v,u}$ as the square root of the number of occurrences of u in the contexts of v, the objective in (1) becomes:

$$\begin{aligned}&\arg\min_{\mathbf{A},\mathbf{B}} \frac{1}{2}\left\|\sqrt{\mathbf{F}}\odot(\mathbf{M}-\mathbf{A}\mathbf{B}^\top)\right\|_F^2\\ &=\arg\min_{\mathbf{A},\mathbf{B}} \frac{1}{2}\sum_{v\in\mathcal{V}}\sum_{u\in\mathcal{V}}\mathbf{F}_{v,u}\Big(\mathbf{M}_{v,u}-\langle\mathbf{A}_{v,:},\mathbf{B}_{u,:}\rangle\Big)^2\\ &=\arg\min_{\mathbf{A},\mathbf{B}} \frac{1}{2}\sum_{\boldsymbol{w}\in\mathcal{W}}\sum_{w_l\in\boldsymbol{w}}\sum_{u\in\mathcal{V}}\Big(\mathbf{M}^{\mathbf{w}}_{w_l,u}-\langle\mathbf{A}_{w_l,:},\mathbf{B}_{u,:}\rangle\Big)^2 \quad (2)\end{aligned}$$

  - $\mathbf{M}_{v,u}$ represents if u appears in the context of v in any walk
  - $\mathbf{F}_{v,u}$ is the number of occurrences of u in the contexts of v

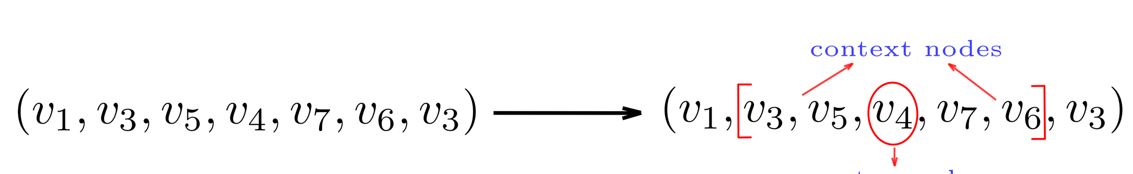

Figure: Illustration of context and center nodes

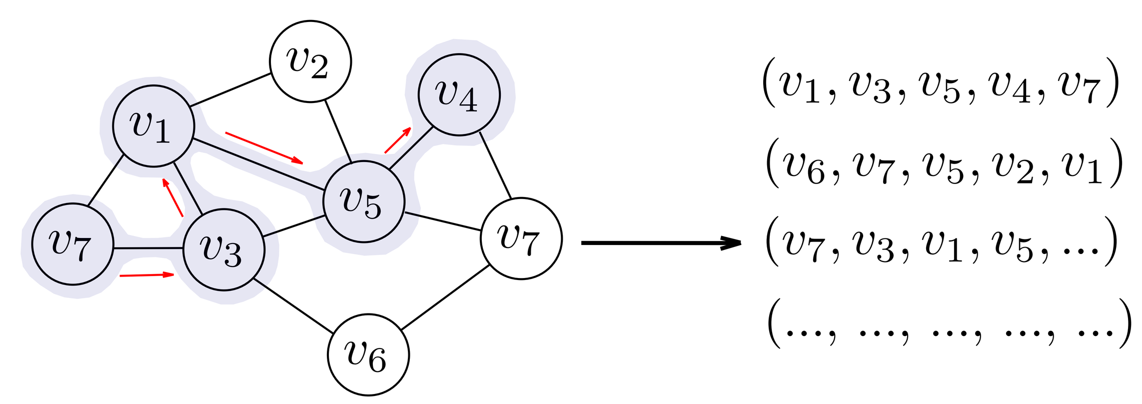

Figure: Generation of random walks

- The inner product in Eq. (2) can be expressed in the feature space as follows:

$$\begin{aligned}&\arg\min_{\mathbf{A},\mathbf{B}} \frac{1}{2}\sum_{\boldsymbol{w}\in\mathcal{W}}\sum_{w_l\in\boldsymbol{w}}\sum_{u\in\mathcal{V}}\Big(\mathbf{M}^{\mathbf{w}}_{w_l,u}-\langle\Phi(\mathbf{A}_{w_l,:}),\Phi(\mathbf{B}_{u,:})\rangle\Big)^2\\ =&\arg\min_{\mathbf{A},\mathbf{B}} \frac{1}{2}\sum_{\boldsymbol{w}\in\mathcal{W}}\sum_{w_l\in\boldsymbol{w}}\sum_{u\in\mathcal{V}}\Big(\mathbf{M}^{\mathbf{w}}_{w_l,u}-\kappa(\mathbf{A}_{w_l,:},\mathbf{B}_{u,:})\Big)^2\end{aligned}$$

- We use the following universal kernels [1, 2] in our evaluation:

$$\kappa_G(x,y)=\exp\left(\frac{-\|x-y\|^2}{\sigma^2}\right) \qquad \kappa_S(x,y)=\frac{1}{\left(1+\|x-y\|^2\right)^{\alpha}}$$

Experimenal Setup

- For optimization, we employ *Stochastic Gradient Descent* (SGD)
- We apply *negative sampling strategy*: k negative instances u^- are sampled from the noise distribution p^- for each context node u^+:

$$\Big(1-\kappa(\mathbf{A}_{v,:},\mathbf{B}_{u^+,:})\Big)^2+\sum_{u^-\sim p^-}\Big(\kappa(\mathbf{A}_{v,:},\mathbf{B}_{u^-,:})\Big)^2.$$

- In experiments, we use logistic regression with L_2 regularization.

	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{K}\rvert$	$\lvert\mathcal{C}\rvert$	**Avg. Degree**	**Type**
CiteSeer	3,312	4,660	6	438	2.814	Citation
Cora	2,708	5,278	7	78	3.898	Citation
DBLP	27,199	66,832	4	2,115	4.914	Co-authorship
AstroPh	17,903	19,7031	-	1	22.010	Collaboration
HepTh	8,638	24,827	-	1	5.7483	Collaboration
Facebook	4,039	88,234	-	1	43.6910	Social
Gnutella	8,104	26,008	-	1	6.4186	Peer-to-peer

Table: Statistics of networks used in the experiments. $\lvert\mathcal{V}\rvert$: number of nodes, $\lvert\mathcal{E}\rvert$: number of edges, $\lvert\mathcal{K}\rvert$: number of labels and $\lvert\mathcal{C}\rvert$: number of connected components.

Numerical Tests

	2%	**4%**	**6%**	**8%**	**10%**	**30%**	**50%**	**70%**	**90%**
DeepWalk	0.416	0.460	0.489	0.505	0.517	0.566	0.584	0.595	0.592
Node2Vec	0.450	0.491	0.517	0.530	0.541	0.585	0.597	0.601	0.599
LINE	0.323	0.387	0.423	0.451	0.466	0.532	0.551	0.560	0.564
HOPE	0.196	0.205	0.210	0.204	0.219	0.256	0.277	0.299	0.320
NetMF	0.451	0.496	0.526	0.540	0.552	0.590	0.603	0.604	0.608
Gauss	0.479	0.514	0.535	0.548	0.560	**0.603**	**0.615**	**0.623**	**0.630**
Sch	**0.482**	**0.519**	**0.538**	**0.552**	**0.561**	0.599	0.613	0.620	0.627

(a) *CiteSeer*

	2%	**4%**	**6%**	**8%**	**10%**	**30%**	**50%**	**70%**	**90%**
DeepWalk	0.545	0.585	0.600	0.608	0.613	0.626	0.628	0.628	0.633
Node2Vec	0.575	0.600	0.611	0.619	0.622	0.636	0.638	0.639	0.639
LINE	0.554	0.580	0.590	0.597	0.603	0.618	0.621	0.623	0.623
HOPE	0.379	0.378	0.379	0.379	0.379	0.379	0.379	0.378	0.380
NetMF	0.577	0.589	0.596	0.601	0.605	0.617	0.620	0.623	0.623
Gauss	**0.611**	**0.621**	**0.626**	**0.628**	**0.630**	**0.637**	**0.641**	**0.642**	**0.644**
Sch	0.610	0.616	0.622	0.624	0.625	0.633	0.636	0.637	0.638

(b) *DBLP*

Table: Micro-F_1 scores for the node classification task

	DeepWalk	Node2Vec	LINE	HOPE	NetMF	Gauss	Sch
CiteSecer	0.837	0.762	0.557	0.756	0.742	**0.886**	0.875
Cora	0.778	0.724	0.554	0.728	0.755	**0.819**	0.814
DBLP	0.944	0.905	0.590	0.930	0.930	**0.963**	0.958
AstroPh	0.960	0.935	0.679	0.967	0.897	**0.978**	0.970
HepTh	0.897	0.830	0.633	0.875	0.882	**0.920**	0.915
Facebook	0.983	**0.988**	0.696	0.980	0.987	0.987	0.987
Gnutella	0.680	0.498	0.702	0.599	0.651	**0.766**	0.677

Table: Area Under Curve (AUC) scores for the link prediction task

Parameter Sensitivity

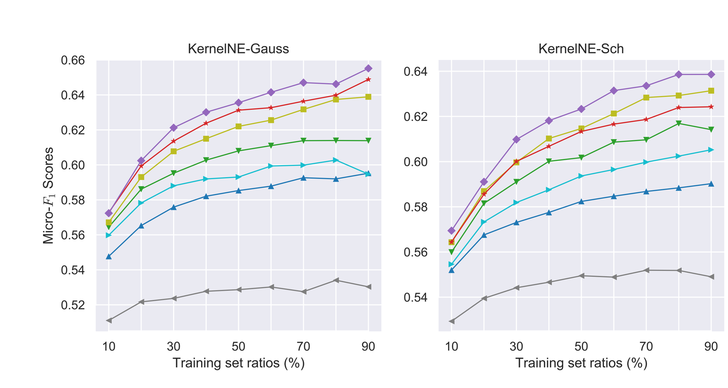

Figure: Dimension size

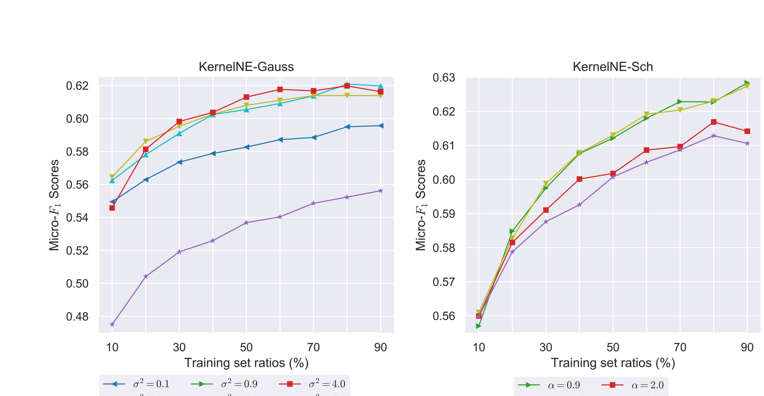

Figure: Kernel parameters

References

[1] C. A. Micchelli *et al.*, "Universal kernels," *J. Mach. Learn. Res.*, vol. 7, 2006.

[2] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines," *J. Mach. Learn. Res.*, vol. 2, 2002.

[3] D. Zhang *et al.*, "Network representation learning: A survey," *IEEE Transactions on Big Data*, 2018.