Supplementary Material for " Improving Textual Network Embedding with Global Attention via Optimal Transport"

Anonymous Author(s)
Affiliation
Address
email

1 Competing models

1.1 Topology only embeddings

Mixed Membership Stochastic Blockmodel (MMB) (Airoldi et al., 2008): a graphical model for relational data, each node randomly select a different "topic" when forming an edge.

DeepWalk (Perozzi et al., 2014): executes truncated random walks on the graph, and by treating nodes as tokens and random walks as natural language sequences, the node embedding are obtained using the SkipGram model (Mikolov et al., 2013).

Node2vec (Grover and Leskovec, 2016): a variant of DeepWalk by executing biased random walks to explore the neighborhood (*e.g.*, Breadth-first or Depth-first sampling).

Large-scale Information Network Embedding (LINE) (Tang et al., 2015): scalable network embedding scheme via maximizing the joint and conditional likelihoods.

1.2 Joint embedding of topology & text

Naive combination (Tu et al., 2017): direct combination of the structure embedding and text embedding that best predicts edges.

Text-Associated DeepWalk (TADW) (Yang et al., 2015): reformulating embedding as a matrix factorization problem, and fused text-embedding into the solution.

Content-Enhanced Network Embedding (CENE) (Sun et al., 2016): treats texts as a special kind of nodes.

Context-Aware Network Embedding (CANE) (Tu et al., 2017): decompose the embedding into context-free and context-dependent part, use mutual attention to address the context-dependent embedding.

Word-Alignment-based Network Embedding (WANE) (Shen et al., 2018): Using fine-grained alignment to improve context-aware embedding.

Diffusion Maps for Textual network Embedding (DMTE) (Zhang et al., 2018): using truncated diffusion maps to improve the context-free part embedding in CANE.

2 Complete Link prediction results on Cora and Hepth

The complete results for Cora and Hepth are listed in Tables 1 and 2. Results from models other than GANE are collected from (Tu et al., 2017; Shen et al., 2018; Zhang et al., 2018). We have also repeated these experiments on our own, the results are consistent with the ones reported. Note that DMTE did not report results on *Hepth* (Zhang et al., 2018) .

%Training Edges	**15%**	**25%**	**35%**	**45%**	**55%**	**65%**	**75%**	**85%**	**95%**
MMB	54.7	57.1	59.5	61.9	64.9	67.8	71.1	72.6	75.9
node2vec	55.9	62.4	66.1	75.0	78.7	81.6	85.9	87.3	88.2
LINE	55.0	58.6	66.4	73.0	77.6	82.8	85.6	88.4	89.3
DeepWalk	56.0	63.0	70.2	75.5	80.1	85.2	85.3	87.8	90.3
Naive combination	72.7	82.0	84.9	87.0	88.7	91.9	92.4	93.9	94.0
TADW	86.6	88.2	90.2	90.8	90.0	93.0	91.0	93.4	92.7
CENE	72.1	86.5	84.6	88.1	89.4	89.2	93.9	95.0	95.9
CANE	86.8	91.5	92.2	93.9	94.6	94.9	95.6	96.6	97.7
DMTE	91.3	93.1	93.7	95.0	96.0	97.1	97.4	98.2	98.8
WANE	91.7	93.3	94.1	95.7	96.2	96.9	97.5	98.2	99.1
GANE-OT	92.0	94.4	95.7	96.6	97.3	97.6	98.6	98.8	99.2
GANE-AP	**94.0**	**96.4**	**97.2**	**97.4**	**98.0**	**98.2**	**98.8**	**99.1**	**99.3**

Table 1: AUC scores for link prediction on the *Cora* dataset.

%Training Edges	**15%**	**25%**	**35%**	**45%**	**55%**	**65%**	**75%**	**85%**	**95%**
MMB	54.6	57.9	57.3	61.6	66.2	68.4	73.6	76.0	80.3
DeepWalk	55.2	66.0	70.0	75.7	81.3	83.3	87.6	88.9	88.0
LINE	53.7	60.4	66.5	73.9	78.5	83.8	87.5	87.7	87.6
node2vec	57.1	63.6	69.9	76.2	84.3	87.3	88.4	89.2	89.2
Naive combination	78.7	82.1	84.7	88.7	88.7	91.8	92.1	92.0	92.7
TADW	87.0	89.5	91.8	90.8	91.1	92.6	93.5	91.9	91.7
CENE	86.2	84.6	89.8	91.2	92.3	91.8	93.2	92.9	93.2
CANE	90.0	91.2	92.0	93.0	94.2	94.6	95.4	95.7	96.3
WANE-*ww*	92.3	94.1	95.7	96.7	97.5	97.5	97.7	98.2	98.7
DMTE	-	-	-	-	-	-	-	-	-
GANE-OT	93.4	96.2	97.0	97.7	97.9	98.0	98.2	98.6	98.8
GANE-AP	**93.8**	**96.4**	**97.3**	**97.9**	**98.1**	**98.2**	**98.4**	**98.7**	**98.9**

Table 2: AUC scores for link prediction on the *Hepth* dataset.

3 Negative sampling approximation

In this section we provide a quick justification for the negative sampling approximation. To this end, we first briefly review *noise contrastive estimation* (NCE) and how it connects to maximal likelihood estimation, then we establish the link to negative sampling. Interested readers are referred to Ruder (2016) for a more thorough discussion on this topic.

Noise contrastive estimation. NCE seeks to learn the parameters of a likelihood model $p_\Theta(u|v)$ by optimizing the following discriminative objective:

$$J(\Theta) = \sum_{u_i \sim p_d} [\log p_\Theta(y=1|u_i, v) - K\mathbb{E}_{\tilde{u}' \sim p_n}[\log p_\Theta(y=0|\tilde{u}, v)]], \tag{1}$$

where y is the label of whether u comes from the data distribution p_d or the tractable noise distribution p_n, and v is the context. Using the Monte Carlo estimator for the second term gives us

$$\hat{J}(\Theta) = \sum_{u_i \sim p_d} [\log p_\Theta(y=1|u_i, v) - \sum_{k=1}^{K} [\log p_\Theta(y=0|\tilde{u}_k, v)]], \tilde{u}_k \overset{iid}{\sim} p_n. \tag{2}$$

Since the goal of $J(\Theta)$ is to predict the label of a sample from a mixture distribution with $1/(K+1)$ from p_d and $K/(K+1)$ from p_n, plugging the model likelihood and noise likelihood into the label likelihood gives us

$$p_\Theta(y=1; u, v) = \frac{p_\Theta(u|v)}{p_\Theta(u|v) + Kp_n(u|v)}, \; p_\Theta(y=0; u, v) = \frac{Kp_n(u|v)}{p_\Theta(u|v) + Kp_n(u|v)}. \tag{3}$$

Recall $p_\Theta(u|v)$ takes the following softmax form

$$p_\Theta(u|v) = \frac{\exp(\langle u, v\rangle)}{Z_v(\Theta)},\ Z_v(\Theta) = \sum_{u'} \exp(\langle u', v\rangle). \tag{4}$$

NCE treats Z_v as an learnable parameter and optimized along with Θ. One key observation is that, in practice, one can safely clamp Z_v to 1, and the NCE learned model ($p_{\hat{\Theta}}$) will *self-normalize* in the sense that $Z_v(\hat{\Theta}) \approx 1$. As such, one can simply plug $p_\Theta(u|v) = \exp(\langle u, v\rangle))$ into the above objective. Another key result is that, as $K \to \infty$, the gradient of NCE objective recovers the gradient of softmax objective $\log p_\Theta(u|v)$ (Dyer, 2014).

Negative sampling as NCE. If we set $K = \#(\mathcal{V})$ and let $p_n(u|v)$ be the uniform distribution on $\mathcal{V}$, we have

$$p_\Theta(y = 1|u, v) = \sigma(\langle u, v\rangle), \tag{5}$$

where $\sigma(z)$ is the sigmoid function. Plugging this back to the $\hat{J}(\Theta)$ covers the negative sampling objective Eqn (6) used in the paper. Combined with the discussion above, we know Eqn (6) provides a valid approximation to the log-likelihood in terms of the gradient directions, when K is sufficiently large. In this study, we use $K = 1$ negative sample for computational efficiency. Using more samples did not significantly improve our results (data not shown).

4 Experiment Setup

We use the same codebase from CANE (Tu et al., 2017)[1]. The implementation is based on TensorFlow, all experiments are exectuted on a single NVIDIA TITAN X GPU. We set embedding dimension to $d = 100$ for all our experiments. To conduct a fair comparison with the baseline models, we follow the experiment setup from Shen et al. (2018). For all experiments, we set word embedding dimension as 100 trained from scratch. We train the model with Adam optimizer and set learning rate $1e - 3$. For GANE-AP model, we use best filte size $1 \times 21 \times 1$ for convolution from our abalation study.

5 Ablation study setup

To test how the n-gram length affect our GANE-AP model performance, we re-run our model with different choices of n-gram length, namely, the window size in convolutional layer. Each experiment is repeated for 10 times and we report the averaged results to eliminate statistical fluctuations.

References

Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. 2008. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(Sep):1981–2014.

Chris Dyer. 2014. Notes on noise contrastive estimation and negative sampling. *arXiv preprint arXiv:1410.8251*.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

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

Sebastian Ruder. 2016. On word embeddings - part 2: Approximating the softmax. `http://ruder.io/word-embeddings-softmax/`.

Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Improved semantic-aware network embedding with fine-grained word alignment. In *EMNLP*.

[1]`https://github.com/thunlp/CANE`

Xiaofei Sun, Jiang Guo, Xiao Ding, and Ting Liu. 2016. A general framework for content-enhanced network representation learning. In *arXiv preprint arXiv:1610.02906*.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale information network embedding. In *WWW*.

Cunchao Tu, Han Liu, Zhiyuan Liu, and Maosong Sun. 2017. CANE: Context-aware network embedding for relation modeling. In *ACL*.

Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Chang. 2015. Network representation learning with rich text information. In *IJCAI*.

Xinyuan Zhang, Yitong Li, Dinghan Shen, and Lawrence Carin. 2018. Diffusion maps for textual network embedding. In *NIPS*.