SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

November 2021, Vol. 64 212104:1–212104:11
https://doi.org/10.1007/s11432-019-2635-8

Sampling informative context nodes for network embedding

Danhao ZHU[1,2], Xin-Yu DAI[1*], Jiajun CHEN[1] & Jie YIN[2]

[1]*Department of Computer Science and Technology, Nanjing University, Nanjing* 210031*, China;*
[2]*Library, Jiangsu Police Institute, Nanjing* 210093*, China*

Received 6 May 2019/Revised 12 July 2019/Accepted 20 August 2019/Published online 12 October 2021

Abstract Several modern network embedding methods learn vector representations from sampled context nodes. The sampling strategies are often carefully designed and controlled by specific parameters that enable them to adapt to different networks. However, the following fundamental question remains: what is the key factor that causes some sampling context results to yield better vectors than others on a certain network? We attempted to answer the question from the perspective of information theory. First, we defined the weighted entropy of the sampled context matrix, which denotes the amount of information it takes. We discovered that context matrices with higher weighted entropy generally produce better vectors. Second, we proposed maximum weighted entropy sampling methods for sampling more informative context nodes; thus, it can be used to produce more informative vectors. Herein, the results of the extensive experiments on the link prediction and node classification tasks confirm the effectiveness of the proposed methods.

Keywords network embedding, representation learning, random walk, entropy, weighted entropy

Citation Zhu D H, Dai X-Y, Chen J J, et al. Sampling informative context nodes for network embedding. Sci China Inf Sci, 2021, 64(11): 212104, https://doi.org/10.1007/s11432-019-2635-8

1 Introduction

Network embedding aims to learn low-dimensional node representations that preserve the property of the original network. The learned vectors can effectively support the downstream network inference tasks, such as node classification [1], anomaly detection [2], community detection [3], and missing link prediction [4].

Further, recent advances in network embedding mainly originated from Deepwalk [5]. The method comprises two phases. The first phase identifies the neighborhood (context) nodes to each source node, where the context nodes are sampled using the random walk method. The second phase maximizes the conditional probability of the observing context nodes to the source node. Because Deepwalk can learn high-quality representations and scale to very large networks, the two-step learning paradigm has become popular in various networks, including homogenous network embedding [6–9], heterogeneous network embedding [10, 11], and attributed network embedding [12].

To capture the diversity of connectivity patterns in different networks, some studies have developed flexible context sampling strategies and achieved promising results. Node2vec [6] proposed a biased random walk strategy for balancing breadth-first and depth-first search of neighborhoods. Ref. [9] proposed an attention model, which can guide the random surfer on "where to attend to" as a function of distance from the source node. These studies biased the walks explicitly or implicitly, to better explore the structure of certain networks. However, the fundamental question remains: what is the key factor that causes some sampling context results to yield better vectors than others on a certain network? Previously, the context sampling procedure was indeed a black box. No indicator could help to judge the quality of the sampling results until the optimization was finished and the vectors were evaluated on the downstream tasks.

* Corresponding author (email: daixinyu@nju.edu.cn)

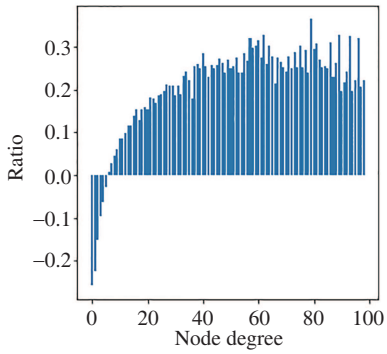

Figure 1 (Color online) Node degree w.r.t. sampled frequency ratio of MWENE/Deepwalk.

To address the question, we assume that the more information the sampled context matrices takes, the better representation will be learned. However, another important question arises: what kind of nodes is informative? Certainly, the context nodes shared by too many source nodes are not very informative. According to information theory, the more an item appears, the less information it takes. However, if a context node appears few times, it does not imply being informative either. Recall the motivation of network embedding and language modeling: entities sharing similar contexts would be similar in vector space. If a context node is shared by a very small number of source nodes, then merely anything would be learned.

From the foregoing, we define the information amount of a sampled context matrix as its weighted entropy. The weighted entropy relates to both the node degree in the original networks and the frequency distribution of the sampled context nodes. Then, we test the context matrices produced using Node2vec and Deepwalk. The learned embeddings are evaluated on link prediction and node classification tasks. The results obtained show that the performances have a significant positive correlation to the weighted entropy.

Based on the findings obtained, we propose a novel neighborhood sampling method named maximum weighted entropy for network embedding (MWENE). MWENE can gradually increase the weighted entropy of the sampled context matrix, and hence can produce more informative vectors. Figure 1 shows the sampling differences between MWENE and Deepwalk, w.r.t. node degrees. MWENE samples much less number of low degree nodes than Deepwalk, as these nodes are less informative in our settings. MWENE samples more nodes as the node degree increases. However, when the node degree becomes extremely large, the speed begins to slow down. From the figure, it is clear that context nodes with middle size of degree will be sampled more. These nodes offer good features for bridging the connection among similar groups and discriminate features for identifying different communities.

Extensive experiments were conducted on link prediction and node classification tasks to determine the effectiveness of the proposed MWENE.

In summary, the contributions of our work are as follows:

• We define weighted entropy to describe the amount of information of the sampled context nodes in network embedding, and validate the fact that context matrices with higher weighted entropy generally yield better representations.

• We present MWENE for sampling high entropy context nodes in network embedding.

• Extensive experiments show that MWENE can outperform the state-of-the-art methods on link prediction and node classification tasks.

The remaining parts of the paper are organized as follows. Section 2 defines the weighted entropy of the sampled context matrix and validates the fact that the indicator significantly correlates with the performance of downstream tasks. Section 3 introduces the proposed MWENE. Section 4 presents the experiments conducted. Section 5 briefly summarizes the related work. Finally, our conclusion of the work is presented in Section 6.

2 Weighted entropy

2.1 Preliminaries

The network embedding problem can be formulated as follows. Given a network $G = (V, E)$, where V denotes the set of nodes and E denotes the set of edges. Each $v_i \in V$ denotes a data object and each $e_{ij} \in E$ denotes a link from v_i to v_j. Each edge e_{ij} can be associated with a weight w_{ij}. For directed networks, there must be an e_{ji} if $e_{ij} \in E$. For unweighted networks, w_{ij} is constant with the value of 1. The goal of network embedding is to learn the embedding matrix $X \in \mathbb{R}^{|V|\times d}$ where row X_i denotes the embedding of v_i and d denotes the dimension of the embedding.

A context matrix $C \in \mathbb{N}^{|V|\times|V|}$ is sampled from the network, which contains the structure regularity of the original network. Each element C_{ij} in the matrix denotes that v_j is sampled as the context node of v_i for C_{ij} times. A couple of methods defined context matrix explicitly or implicitly. Deepwalk [5] obtained short truncated node sequences with random walk and then applied Skip-gram [13] to learn the node representations. Equivalently, C_{ij} denotes the frequency of v_j is within a small distance to v_i in the sequences. Node2vec [6] proposed biased random walk strategy for generating node sequences. They defined the return parameter p and in-out parameter q to interpolate between breadth-first and depth-first search. Ref. [9] linearly combined context distributions over different distances to the source node, to preserve different types of relational information.

Afterward, some optimization methods can be applied to C to learn the final network embeddings such as hierarchical softmax [5], negative sampling [6], and matrix factorization [14].

2.2 Weighted entropy of context matrix

2.2.1 *Definition*

The context matrix C provides all the information the following optimization algorithm can learn from. We assume that the more information C takes, the better the node embeddings that can be learned. We want to define an indicator that can be used to describe the amount of information in the matrix C. First, we transfer the context matrix C to the context distribution P:

$$P = \frac{C}{\sum_i^{|V|}\sum_j^{|V|} C_{ij}}.$$

According to the information theory, the information of the context node v_j is $-\log_2 p_j$ bits, where $p_j = \sum_i^{|V|} P_{ij}$ denotes the probability of v_j in P. However, such a measurement treats all nodes equally and ignores the information quality associated with them. Nodes with a higher degree are generally more important to the network's topology and should be emphasized more. For instance, in a citation network, highly cited papers are often more representative to a certain research area, than the less cited ones. In most real-world networks, the distribution of degree follows the power law. Thus, a general approach is to use a coefficient $\log_2 d_j$ to describe the information quality of context node v_j, where d_j denotes v_j's degree measured from the original network. The weighted information of v_j can thus be represented as

$$\mathrm{WI}(v_j) = -\log_2 d_j \log_2 p_j. \tag{1}$$

Note that some networks may contain nodes with degree 1. The weight information of these nodes will be 0. A practical solution is to let $d_i = 2d_i$ for all nodes when a network contains 1 degree nodes.

The overall weighted information of the context distribution P is the summation of weighted information possessed by each element in the matrix:

$$\mathrm{WI}(C) = \sum_i^{|V|}\sum_j^{|V|} P_{ij}\mathrm{WI}(v_j) = -\sum_j^{|V|} p_j \log_2 d_j \log_2 p_j, \tag{2}$$

where $\mathrm{WI}(C)$ is indeed the weighted entropy proposed in [15]. In this paper, $\mathrm{WI}(C)$ is used to measure the information of C. We assume that the bigger the weighted entropy, the better the vectors may be learned.

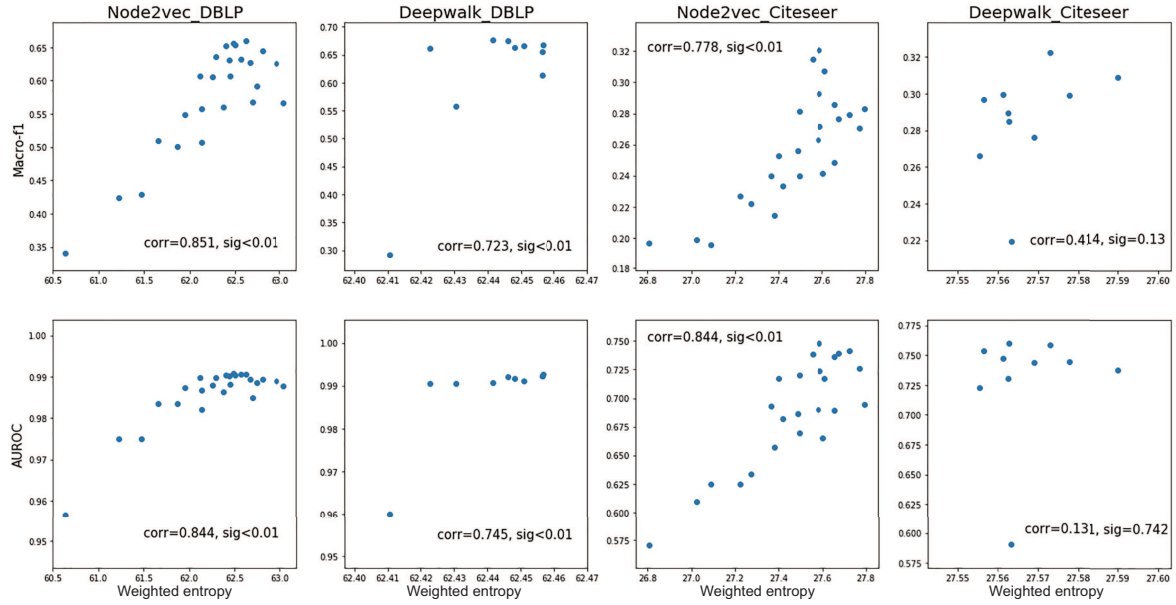

Figure 2 (Color online) Weighted entropy w.r.t. the performance of classification and link prediction tasks. corr and sig denote the correlation coefficient and the significance, respectively.

Table 1 Correlation coefficient between the information measurement and the evaluation indicators. F1 and ROC denote Macro-f1 and AUROC respectively. ** denotes that the significant value p is below 0.01

Information indicator	F1 (DBLP)	ROC (DBLP)	F1 (CITESEER)	ROC (CITESEER)
Entropy	0.132	−0.109	0.397	0.509
KL-divergence	−0.534	0.216	0.163	0.274
Weighted entropy	0.670^{**}	0.660^{**}	0.861^{**}	0.758^{**}

2.2.2 *Validation*

We validate our assumption on node classification and link prediction tasks, with 2 state-of-the-art network sampling techniques: Deepwalk [5] and Node2vec [6]. We exclude [9] in the investigation because their weighted context matrix is obtained automatically. To eliminate the effect of different optimization methods, we use stochastic gradient descent with negative sampling in all the optimization stages. Classification and link prediction tasks are evaluated with Macro-f1 and AUROC, respectively. The datasets used are DBLP and CITESEER. The optimization method, tasks, and datasets will be introduced in detail in the subsequent sections. Particularly, in Deepwalk, window_size was set to 1,2,...,10, respectively. In Node2vec, p, q were set in $[0.25, 0.5, 1, 2, 4]$. In total, we obtained 25 data points for Node2vec and 10 data points for Deepwalk.

The results are shown in Figure 2. Most of the results indicate that weighted entropy has a significant positive correlation with the performance of both tasks. The results on Deepwalk-CITESEER is not significant. This is partly because tuning context size cannot affect the weighted entropy effectively.

We also evaluated some other information measurement indicators, including: (1) entropy of the context matrices; (2) KL-divergence between the sampled frequency distribution and the network degree distribution. We mixed the results of Deepwalk and Node2vec, and reported the correlation existing between the information measurement indicators and the evaluation indicators in Table 1. Weighted entropy has a significant correlation with the evaluation indicators. The indicator of entropy and KL-Divergence is not statistically significant.

In all, the results validate our assumption.

2.3 Discussion

It is important to understand that context matrix with high weighted information often yields better results, though not necessarily. Suppose we sum up a matrix to a row and leave other parts all 0. Such a matrix is of the same weighted entropy as the original one, but we can learn nothing from it. However,

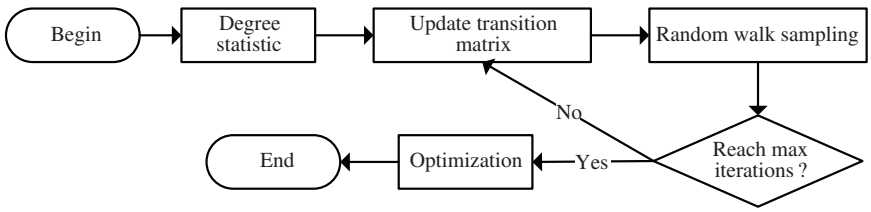

Figure 3 Block diagram of MWENE algorithm.

the context matrix is obtained from the method of random walk. Empirically, the matrices generated by random walk follow this assumption.

A context matrix with large weighted entropy should have the following properties. First, the term $\log d_j$ requires the high-degree nodes to be sampled more than the low-degree nodes. As we discussed earlier, the high-degree nodes are generally more representative and more important to the network. Second, the term $-p_j \log_2 p_j$ requires that the distribution curve of p_j should not be too steep. That is, a major part of nodes should distribute more uniformly. Thus, it is appreciated if each context node is not sampled too much, nor too less. Refer to Figure 1 for more explanation on this, the middle-degree nodes get more sampled opportunity.

Ref. [15] studied the analytical solution of the maximum weighted entropy problem. However, the constrain of the random walk is difficult to model analytically. In the next section, we propose a simple way to iteratively increase the weighted entropy of sampled context nodes.

3 Proposed method

Our learning framework follows the popular two-stage framework proposed in [5], including a context sampling stage and an optimization stage. The context sampling stage takes a network G as input and produces the sampled context matrix C as output. The optimization stage generates the learned embeddings X with C.

3.1 Sampling algorithm: MWENE

For a network G, we want to sample a context matrix that preserves the structural property of the original network. The matrix can be obtained via random walk, while the links are weighted in a transition matrix to enlarge the weighted entropy. The block diagram of the MWENE algorithm is shown in Figure 3. We simulate random walks to obtain the context matrix in each iteration. Then, with the degree of the original network and the context distribution of the context matrix, we can maintain a vector of the weighted information of each node. After that, the weighted information vector can be used to update the transition probability of the original network. Then, the next sampling iteration begins. After the sampling stage, an optimization algorithm can be used to learn the embeddings.

The pseudocode of MWENE is presented in Algorithm 1. The sampling procedure is running for n iterations over all the nodes. The nodes with large weighted information in the previous sampling iterations will be sampled more in the next iteration and vice versa. There are two phases in each iteration. Phase 1 is from line 3 to 13. The random walk starts from each source node in V respectively. Let $g(v_j|v_i)$ be the transition value from v_i to v_j:

$$g(v_j|v_i) = \begin{cases} q_j w_{ij}, & \text{if } e_{ij} \in E, \\ 0, & \text{else}, \end{cases}$$

where q_j denotes the weighted information of v_i and w_{ij} denotes the weight of the original edge in the network. Then, we can save the unnormalized transition value as an array π_{neigh} and normalize it. For each source node, we simulate t steps of random walk to get t context nodes. Phase 2 is lines 14–16, which updates the weight entropy vector after an iteration of sampling finishes.

Apart from the weighted entropy-based sampling, the other advantage of our method is that it can be sued to sample an equal number of context nodes for each source node. Most of the methods proposed in existing works, e.g., [5,6], produce node sequences first, and then use the slide window principle to obtain

Algorithm 1 Maximum weighted entropy sampling for network embedding (MWENE)

Require: Network $G = (V, E)$, walk length t, number of walks per node n;
Ensure: The sampled context matrix $C \in \mathbb{N}^{|V|\times|V|}$.
Initialization: context matrix C where each element is 0, weighted entropy vector $q \in \mathbb{R}^{|V|}$ where each element is 1;
for iteration = 1 to n **do**
  for $i = 1$ to $|V|$ **do**
    $s = i$; //v_s is the current node
    for $j = 1$ to t **do**
      neighbors = get_neighbors(G, s);
      $\pi_{\text{neigh}} = [w_{st}q_t$ for t in neighbors];
      $\pi_{\text{neigh}} = \pi_{\text{neigh}}/\text{sum}(\pi_{\text{neigh}})$;
      $k = \text{AliasSample}(\text{neighbors}, \pi_{\text{neigh}})$; //Sample the next node
      $C_{ik} = C_{ik} + 1$;
      $s = k$;
    end for
  end for
  for $i = 0$ to $|V|$ **do**
    Update q_i according to Eq. (1);
  end for
end for
return C.

the source-context pairs. Because the random walk can merely reach the low-degree nodes, the low-degree nodes would get much less training opportunity than the high-degree nodes. MWENE gives the low-degree nodes a smaller transition weight than the standard random walk. Thus, such an unbalance problem is even more severe. By fixing the number of context nodes for each source node, the proposed method can learn more stable vectors.

3.2 Objective and optimization

Similar to [5,6], we propose to maximize the likelihood of the context nodes from a given ource node. The underlying idea is that the source nodes sharing similar context nodes should be close in vector space. The probability that v_j is the context node of v_i is defined as a softmax function over all the nodes:

$$p(v_j|v_i) = \frac{\mathrm{e}^{X_j^{\mathrm{T}}X_i}}{\sum_{k=1}^{|V|}\mathrm{e}^{X_k^{\mathrm{T}}X_i}}.$$

By assuming conditional independence of the source-context node pairs, the overall objective is as follows:

$$O(X) = \left[\sum_{i}^{|V|}\sum_{j}^{|V|} C_{ij}X_i^{\mathrm{T}}X_j - \log Z\right], \tag{3}$$

where partition function $Z = \sum_i^{|V|}\sum_j^{|V|} X_i^{\mathrm{T}}X_j$ can be estimated with negative sampling [6].

We optimize the objective function in (3) using stochastic gradient descent over the model parameters X. Specifically, we apply the adaptive moment estimation (Adam) [16], which adapts the learning rate according to parameter frequency. In each mini-batch, a random batch of source-context pairs is removed from C and fed to the optimization function, until C is empty.

3.3 Algorithm complexity

The complexity of MWENE is the same as that of previous works such as Deepwalk and Node2vec. The difference is that we have to maintain the weighted information vector after every iteration of sampling, whose complexity is linear to the number of nodes $O(n|V|)$. The additional computation does not affect the overall complexity. To get better parallelization, we do not update the shared weighted information vector after each iteration. Instead, we update it after several iterations, which is encoded in a parallel thread. In practice, it would be hard to observe a significant performance decrease.

Table 2 Statistics of the datasets

Datasets	BlogCatalog	DBLP	CITESEER
$\|V\|$	10312	60744	3312
$\|E\|$	333983	52914	4675
Label count	39	4	6
Label	Interest	Area	Area

4 Experiments

4.1 Experiment setup

4.1.1 *Datasets*

The statistics of the datasets used in this study are shown in Table 2.

• BlogCatalog [17] is a network of social relations provided by blogger authors. The labels represent the topic categories provided by the authors. Each node may have multiple labels in the dataset.

• DBLP[1)] is a citation database. Each paper is labeled with one of the research areas: DB, DM, AI, and CV.

• CITESEER[2)] is a citation database where each paper is labeled with one of the research areas: agents, AI, DB, IR, ML, and HCI.

4.1.2 *Baselines*

We apply the following four network embedding methods as baselines. We implement Deepwalk and Node2vec according to their respective papers. To make a fair comparison, we alter the walk framework of Node2vec the same way as MWENE but keep their biased transition weight strategy. Therefore, the main performance difference between Node2vec and MWENE is purely caused by the context matrices. For GraGep and graph attention, we use their published code. We set the dimension of network embeddings for all methods to 128.

• Deepwalk [5]. It is a pioneering work that introduces random walk and language modeling to network embedding. We set window_size = 10, walk_length = 80, and number_walks = 80.

• Node2vec [6]. The work uses hyper-parameters p and q to simulate a biased random walk. We search $p, q \in \{0.25, 0.5, 1, 2, 4\}$ according to different datasets. Because we change their sampling framework the same as MWENE, the rest of the parameters are similar to that of MWENE.

• GraGep [18]. The method extends high-order proximity and uses the singular value decomposition (SVD) to train the model. The final representation is the concatenation of first-order and high-order vectors. We set kstep = 4.

• Graph attention [9]. The method uses auto learned coefficients to weight context matrices at different distances to the source node. We set $\beta = 0.5$, epoches = 200, $\gamma = 0.5$, and window_size = 10.

4.1.3 *Training details*

Because our proposed biased walking strategy is guided by the predefined weighted information, MWENE does not need to introduce special hyper-parameters such as p, q in Node2vec. Specifically, we set number of walks per node $n = 2000$, walk length $t = 10$, and embedding_size = 128. We initialized the parameters of the matrices randomly with a uniform distribution between $[-1, 1]$, and trained the models with mini-batch Adam [16] with a batch size of 1024. The number of negative sampling is 64.

4.2 Link prediction

4.2.1 *Task description*

The link prediction task aims to predict whether two nodes are linked in the test set, even though they are not linked in the training set. Each dataset of links is divided into a training set and a test set with a training ratio of 0.8. We used normalized Cosine angle to measure the similarity between two vectors, and area under the ROC curve (AUROC) [19] to evaluate the similarities. Because the datasets have only positive edges, we added the same number of random fake links as negative samples to the test set.

1) V4 version. https://www.aminer.cn/citation.

2) http://citeseerx.ist.psu.edu/.

Table 3 Results of link prediction task

Method	BlogCatalog	DBLP	CITESEER
Deepwalk	0.721	0.843	0.796
Node2vec	0.890	0.872	0.815
GraGep	0.758	0.858	0.553
Graph attention	–	**0.904**	0.831
MENE	0.838	0.854	0.802
MWENE	**0.902**	0.875	**0.837**

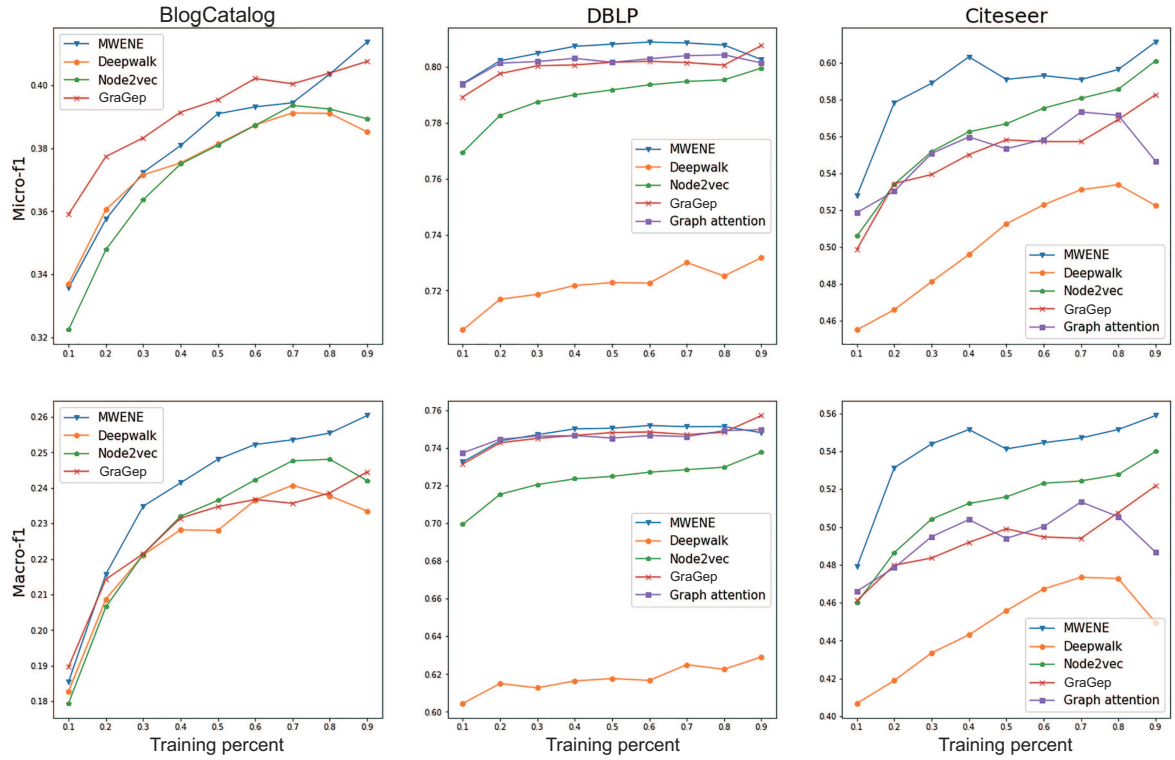

Figure 4 (Color online) Results on node classification task.

4.2.2 *Results*

The results obtained are shown in Table 3. Note that although we test quite a lot of parameter settings, graph attention cannot converge well on Blogcatalog. Hence, we exclude the results on both tasks.

In MWENE, we maximize the entropy instead of the weighted entropy. The performance of MWENE is not good, which gives evidence to the importance of the weighted coefficient of $\log_2 d_j$ in (1).

The method we proposed achieved the best results on Blogcatalog and CITESEER. On DBLP, graph attention showed powerful performance and MWENE got the 2nd best score. We believe that on some networks, the distance of the context-source pairs is a more dominant factor to the link prediction task than weighted entropy. MWENE consistently outperformed Node2vec on all datasets, which indicates the effectiveness of the sampling context matrices with larger weighted entropy.

4.3 Node classification

4.3.1 *Task description*

Node classification aims to predict the label of the node in the test set. Therefore, the task can assess if the learned vectors contain sufficient useful information for the downstream tasks. We trained each network for one epoch to get the embeddings. We randomly sampled training percent by using $\{0.1, 0.2, \ldots, 0.9\}$ embeddings as the training set, and reported the Macro-f1 and Micro-f1 on the rest of the data. For all models, we used simple logistic regression as the classifier. We repeated the experiments for 10 random seed initializations and the results obtained are statistically significant with a p-value of less than 0.01.

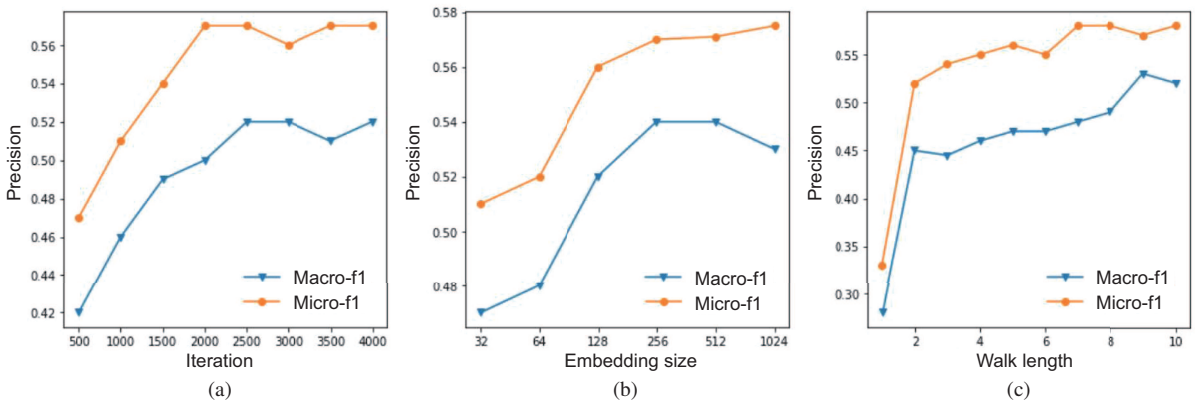

Figure 5 (Color online) Parameter sensitivity. (a) Iteration w.r.t. performance; (b) embedding size w.r.t. performance; (c) walk length w.r.t. performance.

4.3.2 *Results*

The results are shown in Figure 4. Overall, MWENE can achieve better results than other baseline methods. Specifically, on CITESEER with 40% training ratio, the proposed method produces 3.9% improvements on Macro-f1 over Node2vec. On Blogcatalog with 90% training ratio, the proposed methods produced 1.8% improvements on Macro-f1. Although graph attention showed strong performance on link prediction task, however, it did not give outstanding results on node classification task. We believe be distance information may not bet critical in producing informative vectors. Again, MWENE outperformed the Node2vec, which shows that matrices with larger weighted entropy induce more informative vectors.

Overall, the advantage of MWENE is more evident on the Macro-f1 value. We believe this could be attributed to the altered walking strategy: each source node has the same number of context nodes and hence, gets equal training opportunity. Macro-f1 is better for different categories and can benefit from equal training opportunity more.

4.4 Parameter sensitivity

We present the number of iterations n, the walk length t, and embedding size d, against the classification performance on CITESEER. The results are shown in Figure 5. From Figure 5(a), around 2000–2500 iterations can be seen, and the classification precisions are close to the best. From Figure 5(b), it is obvious that the best precision was obtained when the dimension is around 128–256. Figure 5(c) indicates that at up to length 9, 10 of walk, the performance stops increasing significantly. Generally, continuous increase in complexity and training cost may lead to slight gains but is not economical in practice.

4.5 Property analysis

Herein, we investigate some properties of MWENE; as shown in Figure 6. Figure 6(a) shows the weighted entropy of C w.r.t. sample iterations. The weighted entropy converges after a few iterations of sampling. The results show that MWENE can effectively be used to obtain a context matrix with maximum weighted entropy. Figure 6(b) shows the amount of information associated with nodes of different degrees. The properties validate the discussion presented in Section 2. On one hand, the larger the degree, the larger the weighted information. Conversely, the increasing speed slows down as the degree gets large.

5 Related work

Weighted entropy has long been studied since 1971 [15]. It provides a more flexible ability to measure the amount of information than the original definition of entropy. In recent years, it has been employed in various research areas such as evaluating feature importance in clustering [20] and low-cost quantization of deep neural networks [21]. As a flexible and low-cost effective metric indicator, we believe it has great potential in various fields.

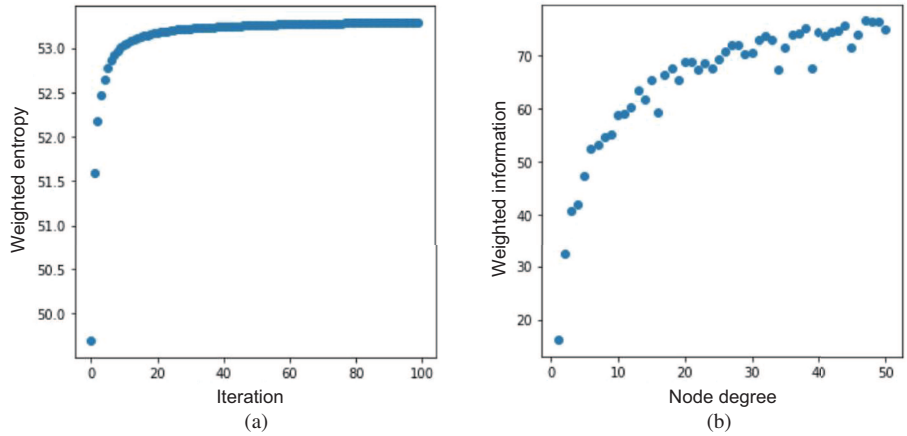

Figure 6 (Color online) Properties of MWENE. (a) Iteration w.r.t. weighted entropy; (b) node degree w.r.t. weighted information.

Deep learning methods show powerful ability in learning features [22, 23]. The most related work to this paper is Deepwalk [5] and Node2vec [6], which we have previously discussed in detail. Another successful embedding method is Line [7], which tried to preserve the first and second-order proximities of networks. There have also been many successful extensions of the random walk method. Ref. [24] used the offsets between vertices observed in a random walk to learn a series of latent representations, each of which captures successively larger relations. Ref. [12] extended Node2vec sampling to attributed network embedding. They modified the encoder of the source node, concatenated the attribute and the structure vectors to obtain the source node vector. Ref. [10] proposed metapath guided random walk strategy for heterogeneous network embedding. Further, they modified the optimization algorithm to adapt to heterogeneous nodes.

We propose weighted entropy to evaluate the quality of different sampled matrices. It is necessary to understand more about the sampling procedure. There may be more accurate and effective indicators to reveal the principle of network embedding.

6 Conclusion

In this paper, we propose weighted entropy for describing the amount of information in sampled context nodes. The results on Deepwalk and Node2vec show that the indicator can effectively represent the quality of learned vectors. Based on the findings obtained, we propose MWENE for sampling more informative context nodes. The sampling method, which is based on the maximum weighted entropy principle, can achieve state-of-the-art results on the evaluation tasks.

Recently, the research interests are shifting to more complex networks such as heterogeneous networks. The node type, link type, and distribution of degree can vary greatly in such a network. Current state-of-the-art networks often require much engineering work such as predefined meta-path. Our future work would focus on the employing weighted entropy idea to heterogeneous network embedding, to find informative context nodes automatically, and reduce the manual works.

Acknowledgements This work was supported in part by Social Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No. 2018SJA0455), National Nature Science Foundation of China (Grant No. 61472183), and Social Science Foundation of Jiangsu Province (Grant No. 19TQD002).

References

1 Sen P, Namata G, Bilgic M, et al. Collective classification in network data. AI Mag, 2008, 29: 93
2 Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. ACM Comput Surv, 2009, 41: 1–58
3 Yang J, McAuley J, Leskovec J. Community detection in networks with node attributes. In: Proceedings of the 13th International Conference on Data Minin, 2013. 1151–1156
4 Liben-Nowell D, Kleinberg J. The link-prediction problem for social networks. J Am Soc Inform Sci Technol, 2007, 58: 1019–1031
5 Perozzi B, Al-Rfou R, Skiena S. Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014. 701–710
6 Grover A, Leskovec J. node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. 855–864

7 Tang J, Qu M, Wang M Z, et al. Line: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, 2015. 1067–1077
8 Wang D X, Cui P, Zhu W W. Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. 1225–1234
9 Abu-El-Haija S, Perozzi B, Al-Rfou R, et al. Watch your step: learning node embeddings via graph attention. In: Proceedings of the 32nd Conference on Neural Information Processing System, 2018. 9198–9208
10 Dong Y X, Chawla N V, Swami A. metapath2vec: scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017. 135–144
11 Zhang D K, Yin J, Zhu X Q, et al. MetaGraph2Vec: complex semantic path augmented heterogeneous network embedding. In: Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2018. 196–208
12 Liao L Z, He X N, Zhang H W, et al. Attributed social network embedding. IEEE Trans Knowl Data Eng, 2018, 30: 2257–2270
13 Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. 2013. ArXiv:1301.3781
14 Qiu J Z, Dong Y X, Ma H, et al. Network embedding as matrix factorization: unifying deepwalk, line, pte, and node2vec. In: Proceedings of the 11th ACM International Conference on Web Search and Data Mining, 2018. 459–467
15 Guiaşu S. Weighted entropy. Rep Math Phys, 1971, 2: 165–179
16 Kingma D P, Ba J. Adam: a method for stochastic optimization. 2014. ArXiv:1412.6980
17 Tang L, Liu H. Relational learning via latent social dimensions. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009. 817–826
18 Cao S S, Lu W, Xu Q K. Grarep: learning graph representations with global structural information. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015. 891–900
19 Zou K H, O'Malley A J, Mauri L. Receiver-operating characteristic analysis for evaluating diagnostic tests and predictive models. Circulation, 2007, 115: 654–657
20 Yang M S, Nataliani Y. A feature-reduction fuzzy clustering algorithm based on feature-weighted entropy. IEEE Trans Fuzzy Syst, 2018, 26: 817–835
21 Park E, Ahn J, Yoo S. Weighted-entropy-based quantization for deep neural networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017
22 Feng X C, Qin B, Liu T. A language-independent neural network for event detection. Sci China Inf Sci, 2018, 61: 092106
23 Li X L, Zhuang Y, Fu Y J, et al. A trust-aware random walk model for return propensity estimation and consumer anomaly scoring in online shopping. Sci China Inf Sci, 2019, 62: 052101
24 Perozzi B, Kulkarni V, Skiena S. Walklets: multiscale graph embeddings for interpretable network classification. 2016. ArXiv:1605.02115