Improving Skip-Gram based Graph Embeddings via Centrality-Weighted Sampling

Pedro Almagro Blanco PEDRO.ALMAGRO@UB.EDU
Universitat de Barcelona Institute of Complex Systems (UBICS)
Departament de Física de la Matèria Condensada
Universitat de Barcelona, Martí Franquès 1, E-08028 Barcelona, Spain.
Complex Systems Modelling Group
Universidad Central de Ecuador
Quito, 170129, Ecuador.

Fernando Sancho Caparrini FSANCHO@US.ES
Dpt. Computer Science and Artificial Intelligence
University of Sevilla
Sevilla, 41012, Spain.

Abstract

Network embedding techniques inspired by *word2vec* represent an effective unsupervised relational learning model. Commonly, by means of a *Skip-Gram* procedure, these techniques learn low dimensional vector representations of the nodes in a graph by sampling *node-context* examples. Although many ways of sampling the *context* of a node have been proposed, the effects of the way a *node* is chosen have not been analyzed in depth. To fill this gap, we have re-implemented the main four *word2vec* inspired graph embedding techniques under the same framework and analyzed how different sampling distributions affects embeddings performance when tested in node classification problems. We present a set of experiments on different well known real data sets that show how the use of popular centrality distributions in sampling leads to improvements, obtaining speeds of up to 2 times in learning times and increasing accuracy in all cases.

1. Introduction

The application of neural encoders to texts has provided very interesting results. In 2013, T. Mikolov et al. (Mikolov, Chen, Corrado, & Dean, 2013a) presented two architectures, under the generic name of *word2vec*, minimizing computational complexity of word representation while maintaining grammatical properties present in the texts from which the words are extracted: Continuous bag-of-words (CBOW), and Skip-gram. In them a *context of a word* in a text is defined as the set of words that appear in its adjacent positions, and both architectures consist of feed-fordward artificial neural networks with three layers: an input layer, a hidden (encoding) layer and an output layer, but they differ in the objective function they try to approximate. On the one hand, CBOW architecture receives the context of a given word as input and tries to predict that word as output. On the other hand, Skip-gram architecture receives the word as input and tries to predict the context associated with it. The main objective of the work of Mikolov et al. is to reduce the complexity in the neural model allowing the system to learn from a large volume of textual data. Through the relationship established between vocabulary words and their contexts, the model captures

different types of similarity, both functional and structural, and provides an embedding of words in vector space that reflects these similarities (Mikolov, Yih, & Zweig, 2013d). In the case of Skip-Gram architecture, two main optimizations have been presented: Negative Sampling, that modifies only some of the weights related to negative examples (words not in *context*), and Hierarchical Softmax, where the output vector is determined by a tree-like traversal of the network layers (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013b).

Graph data appears in a number of application domains such as chemistry, social sciences, and physics, where logical problems commonly addressed are node classification (Neville & Jensen, 2000), link prediction (Liben-Nowell & Kleinberg, 2007), and network representation learning (Zhang, Yin, Zhu, & Zhang, 2018), to name but a few. Many successful methods inspired by language modeling have been developed recently for graph embedding. Those methods allow to obtain vector representations of nodes in a low dimensional space through sampling the relations between them in a graph.

In this work, we evaluate the use of centrality measures to improve efficiency of four of the most popular *word2vec* inspired graph embedding techniques: DeepWalk (Perozzi, Al-Rfou, & Skiena, 2014), LINE (Tang, Qu, Wang, Zhang, Yan, & Mei, 2015), node2vec (Grover & Leskovec, 2016) and Neighborhood Based Node Embeddings (NBNE) (Pimentel, Veloso, & Ziviani, 2018). We analyze the previous four models using five different centrality measures, and we obtain some important conclusions: (1) in all cases, centrality-weighted sampling speeds up convergence (x2 in some cases) of node classification tasks and, (2) there is a fixed ranking in the goodness of centralities when used in this context. This conclusions have been obtained from comprehensive experiments over two well-known datasets. Obtained results present a new application for node centrality measures to improve the efficiency of language modeling based graph embedding techniques.

The rest of the paper is arranged as follows. In Section 2 we summarize the fundamentals of language modeling through Skip-Gram model. The four graph embedding techniques under evaluation will be presented in Section 3. In Section 4 we present our approach to improve efficiency of Skip-Gram based graph embedding techniques. Section 5 is devoted to outline our experiments, and their results are presented in Section 6. In Section 7 we present works related to this one presented here. We close with our conclusions in Section 8.

2. Language Modeling with Skip-Gram

In the following, we will present some basics related to language modeling necessary to describe the implementation that we have carried out of the four methods under study. In general terms, the goal of language modeling is to estimate the likelihood of a specific sequence of words appearing in a corpus. More formally, given a sequence of words $w_1, w_2, ..., w_n$, where $w_i \in \mathcal{W}$ ($\mathcal{W}$ is the vocabulary), we would like to maximize:

$$\Pr(w_n|w_1, w_2, ..., w_{n-1})$$

over all the training corpus.

As mentioned above, recent works have focused on using probabilistic neural networks to build general representations of words. The goal is to learn a latent representation,

$\phi : \mathcal{W} \rightarrow \mathbb{R}^{|\mathcal{W}| \times d}$. This mapping ϕ represents the d-dimensional latent representation associated with each word w in the vocabulary.

Skip-Gram is a language model that, from a corpus of texts, maximizes the co-occurrence probability among the words that appear within a context (of prefixed size, c) in a sentence of the corpus. First, instead of using the context to predict a missing word, it uses one word to predict the context. Secondly, the context is composed of the words appearing to both the right and left of the given word in the sentence. Finally, it removes the ordering constraint on the problem, requiring the model to maximize the probability of any word appearing in the context regardless of its offset from the given word. More formally, given a sequence of training words $w_1, w_2, ..., w_n$, the objective of the Skip-Gram model is to maximize the average log probability:

$$\frac{1}{n} \sum_{i=1}^{n} \sum_{-c \leq j \leq c, j \neq 0} \log \Pr(w_{i+j} \mid w_i)$$

As we will see later, these relaxations are particularly desirable for graph representation learning: the order independence assumption better captures the sense of *neighbourhood* as it is provided in graphs; moreover, this fact will be quite useful for speeding up the training time by building small models giving one vertex at a time.

The basic Skip-Gram formulation defines $\Pr(w_{i+j} \mid w_i)$ using the soft-max function as:

$$\Pr(w_{i+j} \mid w_i) = \frac{\exp(\phi(w_{i+j})'^T \phi(w_i))}{\sum_{w=1}^{W} \exp(\phi(w)'^T \phi(w_i))}$$

where $\phi(w)$ and $\phi(w)'$ are the *input* and *output* vector representations of w, respectively. Optimizing this model by gradient descent means taking a training example and adjusting all the parameters of the model. In other words, each training example will tweak all of the parameters in the model. Usually, the size of the vocabulary means that Skip-Gram model has a tremendous number of parameters, all of which would be updated by every one of the many training examples. To speed up the training time, Skip-Gram authors presented two approximations: Negative Sampling and Hierarchical Soft-max (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013c).

2.1 Negative Sampling

Negative Sampling faces the cost of calculating $\Pr(w_{i+j} \mid w_i)$ by allowing each training example to only modify a small percentage of the parameters, rather than all of them. For each observed pair (w_{i+j}, w_i) we sample k *negative* context words $w \in \mathcal{W}$ from a noise distribution $P_n(w)$.

$$\log \sigma(\phi(w_{i+j})'^T \phi(w_i)) - \sum_{p=1}^{k} E_{w_p} \sim P_n(w)[\log \sigma(\phi(w_p)'^T \phi(w_i))]$$

which is used to replace every $\log \Pr(w_{i+j} \mid w_i)$ term in the Skip-Gram objective. Thus the task is to distinguish the correct samples from negative ones obtained from the noise distribution $P_n(w)$, where there are k negative samples for each positive data sample.

2.2 Hierarchical Soft-max

Other way to deal with the high costs of calculating $\Pr(w_{i+j} \mid w_i)$ is Hierarchical Softmax. This model factorizes the conditional probability assigning the words to the leaves of a binary tree, turning the prediction problem into maximizing the probability of a specific path in the hierarchy. If the path to word w_{i+j} is identified by a sequence of tree nodes, $(b_0 = \text{root},\ b_1,\ b_2,\ ...,\ b_{log|\mathcal{W}|=w_{i+j}})$, then:

$$\Pr(w_{i+j} \mid w_i) = \prod_{l=1}^{log|\mathcal{W}|} \Pr(b_l|w_i)$$

Now, $\Pr(b_l|w_i)$ could be modeled by a binary classifier that is assigned to the parent of the node b_l as next equation shows:

$$\Pr(b_l \mid w_i) = \sigma(\phi(w_i)^T \alpha(b_l))$$

where $\sigma(x) = 1/(1+\exp(-x))$ and $\alpha(b_l) \in \mathbb{R}^d$ is the representation assigned to tree node b_l 's parent. This reduces the computational complexity of calculating $\Pr(w_{i+j}|w_i)$ from $O(|\mathcal{W}|)$ to $O(log|\mathcal{W}|)$. Also, unlike the standard soft-max formulation of the Skip-Gram which assigns two representations $\phi(w)$ and $\phi'(w)$ to each word w, the hierarchical soft-max formulation has one representation $\phi(w)$ for each word w and one representation $\alpha(b_l)$ for every inner parent node of b_l in the binary tree.

3. Graph Embedding Techniques Inspired by Language Modeling

Recent work in graph embedding uses Skip-Gram probabilistic neural networks to build general representations of the nodes in a graph. In this work we analyze how centrality measures can be used to improve efficiency of the four graph embedding techniques presented below. As we don't work with weights nor attributes, we will not describe details of the techniques related with that aspects.

3.1 DeepWalk

DeepWalk is a generalization of language modeling that explores a graph $G = (V, E)$ through a stream of short random walks and learns representation of nodes using a Skip-Gram architecture (Perozzi et al., 2014). These walks can be thought of as short sentences and phrases in a special language; the direct analog is to estimate, given a random walk $v_1, v_2, v_3, ..., v_n$, the likelihood of observing vertex v_{i+j} given a vertex v_i in the random walk, i.e.

$$\Pr(v_{i+j}|v_i) = \frac{\exp(\psi(v_{i+j})'^T \psi(v_i))}{\sum_{v=1}^{V} \exp(\phi(v)'^T \phi(v_i))}$$

The goal is to learn a latent representation, $\psi : V \rightarrow \mathbb{R}^{|V| \times d}$. This mapping ψ represents the d-dimensional latent representation associated with each vertex v in the graph. The objective of the Skip-gram model is to maximize the average log probability:

$$\frac{1}{n} \sum_{i=1}^{n} \sum_{-c \leq j \leq c, j \neq 0} \log \Pr(v_{i+j} \mid v_i)$$

where c represents context size. DeepWalk applies a Skip-Gram model with Hierarchical Soft-max to random walks formulating a method which generates low-dimensional representations of networks in a continuous vector space.

3.2 LINE

LINE (Large-scale Information Network Embedding) also uses a similar Skip-Gram architecture to embed the networks but it differs in some aspects from DeepWalk. First, LINE don't use random walks to generate examples, however it uses solely the neighbor information of every node to preserving both the first-order and second-order proximity. LINE model trains the first-order proximity and second-order proximity separately and then concatenate the obtained embeddings for each vertex (Tang et al., 2015). The first-order proximity tries to maximize for each edge $(v_i, v_j) \in E$, the joint probability:

$$\Pr(v_i \mid v_j) = \sigma(\psi(v_i)^T \psi(v_j))$$

As it uses the same *input* and *output* vector representations, first-order proximity can deal only with undirected graphs. To avoid the trivial solution $\psi(v_{ik}) = \infty$, for $i = 1, ..., |V|$ and $k = 1, ..., d$, the first-order proximity is approximated using a Negative Sampling architecture. The second-order proximity is also learned using Negative Sampling but using different *input* and *output* vector representations (as presented in Section 2.1). In both cases positive examples are formed by a node and one of his neighbors and negative examples are extracted from a modified unigram distribution $P_n(v) \propto d_v^{3/4}$ as proposed in (Mikolov et al., 2013c), where d_v is the out-degree of vertex v.

3.3 node2vec

node2vec is a generalization of DeepWalk in which random walks are guided by two parameters p and q. Given a random walk that has just crossed the edge $(t, v) \in E$, the probability of taking the edge $(v, x) \in E$ corresponds to:

$$\gamma_{pq}(v, x) = \begin{cases} \frac{1}{p} & \text{if } dis(t, x) = 0 \\ 1 & \text{if } dis(t, x) = 1 \\ \frac{1}{q} & \text{if } dis(t, x) = 2 \end{cases}$$

where $dis(t, x)$ represents the length of the minimum path between node t and node x. node2vec is identical to DeepWalk with the exception that it explores new methods to generate random walks, at the cost of introducing more hyperparamenters (Grover & Leskovec, 2016).

3.4 Neighborhood Based Node Embedding

Neighborhood Based Node Embedding (NBNE) uses a Skip-Gram model with Negative Sampling to learn representations of nodes in a graph from positive examples formed by a node and one of his neighbors and negative examples extracted from a modified unigram distribution $P_n(v) \propto d_v^{3/4}$. NBNE is equivalent to second-order proximity in LINE, its performance is lower than previous models but its training is faster due to the simplicity of its algorithm (Pimentel et al., 2018).

4. Method

Under the premise that more central nodes are more informative when learning representations for later classification, we have implemented the four methods under the same Negative Sampling architecture and we have introduced λ_i (the prestige of vertex v_i) in the conditional probability.

4.1 Overview

All graph embedding techniques under analysis requires a set of positive examples and a set of negative examples. Negative examples will be drawn from the uniform distribution. Positive examples will be generated in a different manner for each case: DeepWalk and node2vec consider a set of short truncated random walks as corpus from which to extract positive examples while LINE and NBNE generate positive examples using the neighbors of a node.

4.2 Algorithm

To increase efficiency, we decide to use one negative example for each positive example ($k = 1$). Then, in our framework, the term $\log \Pr(v_{i+j} \mid v_i)$ for DeepWalk and node2vec objective function becomes:

$$\log \lambda_i \sigma(\psi(v_{i+j})'^T \psi(v_i)) - E_{v_p} \sim P_n(v)[\log \sigma(\psi(v_p)'^T \psi(v_i))]$$

In the case of DeepWalk, v_{i+j} is obtained from random walks with a context of size $c = 30$. In the case of node2vec, v_{i+j} is obtained from modified random walks as described in Section 3.3 and also with a context of size $c = 30$. The term $\log \Pr(v_j, v_i)$ to be maximized in NBNE becomes

$$\log \lambda_i \sigma(\psi(v_j)'^T \psi(v_i)) - E_{v_p} \sim P_n(v)[\log \sigma(\psi(v_p)'^T \psi(v_i))]$$

for each edge $(v_i, v_j) \in E$. In the case of LINE, we have approximated the *first-order proximity* by using NBNE and *second-order proximity* by selecting nodes at distance 2 as positive examples. The presented framework allows to compare different embedding approaches under a common framework taking advantage of high parallelism.

4.3 Centrality-weighted Sampling

Next we describe the centrality-weighted sampling treatment which improves the effectiveness of the embeddings when facing node classification tasks. We will use centrality information of node v_i to determine parameter λ_i. Centrality measures under consideration are Degree, Betweenness Centrality (BC) (Freeman, 1977), Closeness Centrality (Clos) (Bavelas, 1950), PageRank (PR) (Page, Brin, Motwani, & Winograd, 1998) and Load Centrality (Brandes, 2008). As we will show in next sections, efficiency of all techniques under analysis is improved using proposed sampling techniques.

Dataset	$\|V\|$	$\|E\|$	#classes
Cora	2,708	5,429	7
Citeseer	3,327	4,732	6

Table 1: Dataset statistics.

5. Experimental Design

To train the models, we used 1M positive examples (and 1M negative examples) and 200 dimensions (as in original methods). A single learning process iterates through all positive and negative examples with a batch size of 100k. We used the same random initial weights for every centrality measure following (Grover & Leskovec, 2016). Stochastic gradient descent (SGD) (Recht, Re, Wright, & Niu, 2011) is used in our experiments to optimize free parameters. The derivatives are estimated using the back-propagation algorithm and the learning rate for SGD is 0.1%. Our framework was implemented with the *TensorFlow* (Abadi, Barham, Chen, Chen, Davis, Dean, Devin, Ghemawat, Irving, Isard, Kudlur, Levenberg, Monga, Moore, Murray, Steiner, Tucker, Vasudevan, Warden, Wicke, Yu, & Zheng, 2016) wrapper *Keras* (Chollet et al., 2015). We used the logistic regression classifier from *LibLinear* (Fan, Chang, Hsieh, Wang, & Lin, 2008). All experiments were run on hardware with 32GB RAM, a single 3.4 GHz CPU, and two GeForce GTX 1080 GPUs.

5.1 Datasets

We evaluate the influence of centrality weighted sampling with the following two commonly used benchmark data sets: Cora (McCallum, 2017) and Citeseer (Giles, Bollacker, & Lawrence, 1998). Cora and Citeseer are citation networks where nodes represent documents and links represent citations. In both cases, class labels represent the main topic of the document each node have exactly one class label. Statistics about data sets are summarized in Table 1.

6. Experiments

Let us perform empirical evaluations with the objective of analyze how centrality measures can help to improve the efficiency of presented graph embedding methods when facing node classification.

In the label classification setting, every node is assigned one label from a finite set. During the training phase, we observe the representation of a certain fraction of nodes and their labels. The task is to predict the labels for the remaining nodes. Each experiment has been repeated 4 times, obtaining a standard deviation smaller than 0,006% in all cases. In Figure 6 we show two 2-D representations of node representations achieved with node2vec over CiteSeer data (color of a node indicates the topic of the document). Embedding at the right side have been achieved with Betweenness Centrality weighted sampling and embedding at the left side without weighted sampling. Both embeddings have been achieved training only with 400k samples.

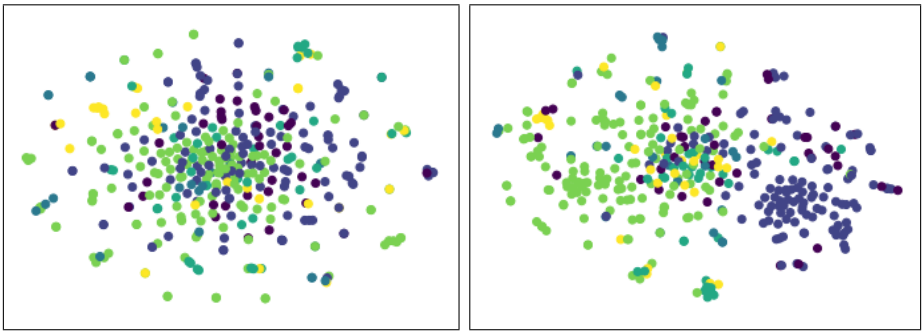

Figure 1: Visualization of a fraction of *CiteSeer* network. Documents are mapped to the 2-D space using the t-SNE package (only 400k samples). (right) Embedding achieved with node2vec+BC, embedding achieved with node2vec baseline.

Metric	DeepWalk	LINE	node2vec	NBNE
BC	**0.6338**	**0.5386**	**0.6337**	**0.5354**
Load	0.6329	0.5375	0.6330	0.5354
Deg	0.6380	0.5206	0.6375	0.5195
PR	0.6330	0.5162	0.6373	0.5116
Clos	0.5932	0.4912	0.5858	0.4876
Base	0.5805	0.4848	0.5952	0.4667

Table 2: *Micro-F1* for node classification in *Cora* data set.

6.1 Cora

Experiments with Cora data set reveal that the use of centrality measures when sampling nodes to participates as positive examples in the different graph embedding methods presented in Section 3 leads to a representative speedup in the convergence to an optimal embedding for node classification.

The results for this data set are listed in Table 2. We include the baseline and the centrality-weighted results. node2vec and DeepWalk significantly outperforms all existing approaches on Cora data set. Both obtain the highest improvement with BC but their results are really similar to those obtained using Load. Also, we can mention that node2vec and DeepWalk do not take full advantage of PageRank centrality. Another fact that we can observe in this table is the clear ranking between different centrality measures. Betweenness Centrality ranks first, next Load, next Degree, next PageRank and at the last place we find Closeness Centrality. Betweenness and Load have similar behaviors. It can be due to the similarity between those two metrics.

Figure 2 presents the relation bwetween Micro-F1 metric and the number of examples used to obtain the embedding in Cora data set for baseline methods and using centrality-weighted samplings. As we can observe, the highest improvement is produced between 200k and 600k samples for the Cora data set. In addition, we confirm really similar behavior for Betweenness Centrality and Load Centrality, for Degree and PageRank and for baseline and Closeness Centrality.

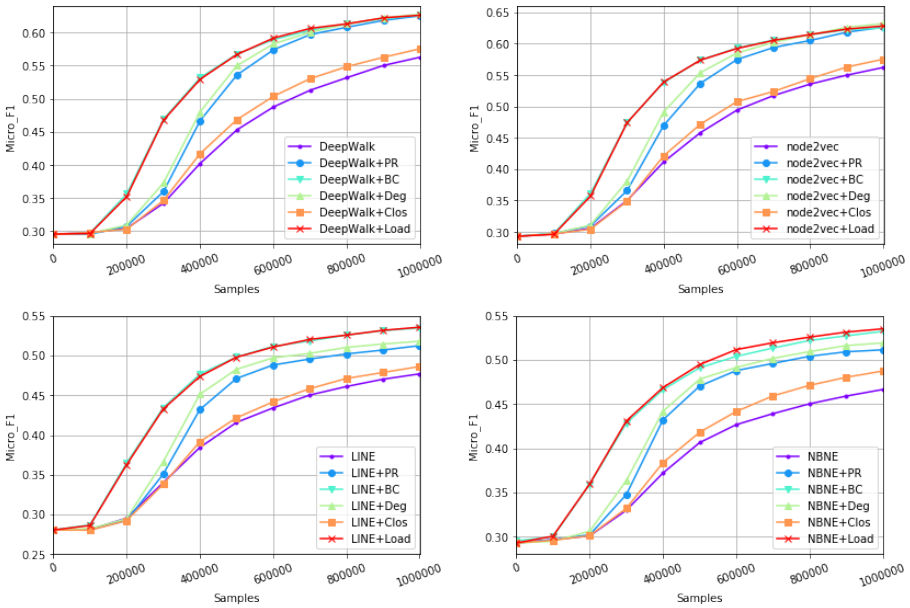

Figure 2: *Micro-F1* for Cora label classification problem using baselines and centrality weighted sampling models.

Metric	DeepWalk	LINE	node2vec	NBNE
BC	0.4525	**0.4222**	**0.4556**	0.4240
Load	**0.4536**	0.4215	0.4542	**0.4249**
Deg	0.4468	0.3977	0.4483	0.4001
PR	0.4448	0.3911	0.4398	0.3929
Clos	0.4234	0.3815	0.4280	0.3863
Base	0.4019	0.3653	0.4061	0.3616

Table 3: *Micro-F1* for node classification in *Citeseer* data set.

6.2 Citeseer

Experiments with Citeseer data set reveal similar insights. Again, the use of centrality measures when sampling nodes to participate as positive examples in the different graph embedding methods presented in Section 3 leads to a representative speedup in the convergence to an optimal embedding for node classification.

The results for Citeseer are listed in Table 3. node2vec and DeepWalk significantly outperforms all presented models on Citeseer classification problem. In this case, node2vec has highest improvement with BC but DeepWalk with Load. node2vec and DeepWalk take advantage of PageRank on CiteSeer data set. The centrality ranking remains so similar than in the Cora case.

Figure 3 present the relation bwetween Micro-F1 metric and the number of examples used to obtain the embedding of the different base models and their centrality weighted

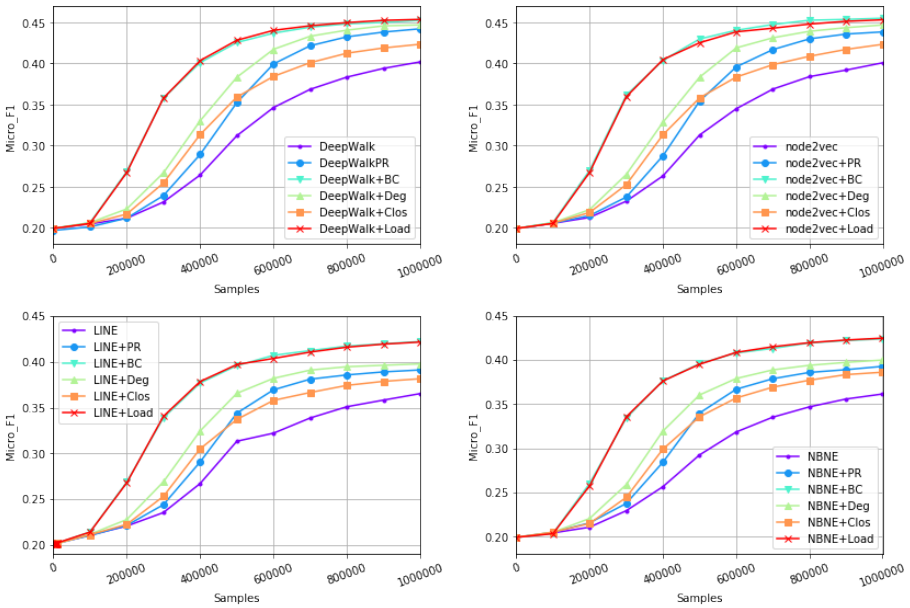

Figure 3: Micro-F1 for Citeseer label classification problem using different centrality measures when sampling and baseline models.

versions for Citeseer data set. As we can observe, the highest improvement is produced again between 200k and 600k samples. We confirm almost the same behavior for Betweenness and Load Centrality in all methods. For node2vec model, Load Centrality performs a little bit better for a big number of examples (close to one million). In this case, Closeness Centrality performs better than on Cora, performing better than PageRank version at the very initial phase (under 500k examples).

7. Related Works

Usually, formalizations of graphs include only positive relation instances, leaving the door open for a variety of methods for selecting negative examples. In (Kotnis & Nastase, 2017) the authors present an empirical study on the impact of Negative Sampling on the learned embeddings, assessed through the task of link prediction. They focus in compare well known methods for Negative Sampling (random generation and corrupting positive examples) and propose two new embedding based sampling methods.

In the work titled *Robust negative sampling for network embedding* the authors provide theoretical arguments that reveal how Negative Sampling can fail to properly estimate the Skip-Gram objective, and why it is not a suitable candidate for the network embedding problem (Armandpour, Ding, Huang, & Hu, 2019). They show that Negative Sampling can learn undesirable embeddings, as the result of the *Popular Neighbor Problem*. This deviation of Negative Sampling from that ideal behavior is mainly caused by allowing a node to choose its neighbor as a negative sample. This problem is more severe when high-

degree nodes are present. They present a new method that alleviates this problem by using a new negative sampling scheme and penalization of the embeddings.

In (Wang, Zhang, Feng, & Chen, 2014) the authors present an optimization of Negative Sampling in the case of embeddings working with relational databases (Bordes, Usunier, Garcia-Duran, Weston, & Yakhnenko, 2013; Wang et al., 2014). They tend to give more chance to replacing the head entity of a relation if it is one-to-many and more chance to replacing the tail entity if the relation is many-to-one. In this way, the chance of generating false negative labels is reduced.

As we shown, some work regarding to the methods selecting negative samples in Skip-Gram graph embedding techniques have been presented. Only LINE model introduces a parameter in the objective function to guide positive examples selection. But LINE authors did not make any analysis of the influence of this parameter in the efficiency of the method and did not present any advantage of this approach. Our work fills this gap presenting a detailed study of such influence not only over LINE but over other similar graph embedding techniques.

8. Conclusions

The use of probability distributions in the selection of positive examples when using graph embedding methods based on Skip-Gram allows to obtain a higher performance in node classification tasks. Both the methods based on random paths and those that construct the positive examples only from the vicinity of the nodes have demonstrated a significant improvement in efficiency by using distributions related to centrality measures in the nodes of the graph. From our knowledge, this work represents the first analysis of this Skip-Gram modification on graphs. Experiments on real data illustrate the effectiveness of our approach on challenging label classification tasks. Our results show that we can create fast and scalable meaningful representations for large graphs making use of centrality measures when selecting positive examples. Our method significantly outperforms other methods designed for the same purpose.

Starting from the premise that the centrality of a node in a network is a sign that this node is more informative when performing an embedding, in this work we have presented an analysis concerning some of the most popular measures of centrality. The results show a significant improvement in all methods under study. Specifically, centralities not based on random paths such as Betweenness Centrality have demonstrated their usefulness to accelerate embedding proceses. In addition, the results show a clear ranking in the centralities according to their goodness. Experiments with a bigger number of examples and with other centrality measures, datasets and methods are still pending. Experiments regarding other rekational learning tasks as link prediction should be also considered. Another worth to mention future line of research consists on applying centrality-based distributions to negative examples.

One of the most interesting conclusions that can be drawn from this study is derived from the fact that Betweenness and Load centralities are the most successful measures in the presented context. From our point of view, this is because the information that these two measures provide is different from the information contained in the random paths and in the first and second order proximities in which models under study are based. Both centralities

contain information on minimum paths not explicitly explored by methods under analysis and we believe that this fact is key to understand the improvement presented.

Acknowledgments

The work was supported in part by the GLVEZUS Project of Universidad Central de Ecuador, by the project "Metodologías de datos aplicadas al análisis de las exposiciones artísticas en Andalucía para el desarrollo de la economía creativa", from Fundación Pública Centro de Estudios Andaluces (2017-2019) and by the projects TIN2013-41086-P (LOCOCIDA Project) and FIS2016-76830-C2-2-P (Adaptabilidad y Cooperacion en Sistemas Biosociales en la Multiescala II) from Ministerio de Economía, Industria y Competitivad of Spain (FEDER funds from EU).

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., & Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283.

Armandpour, M., Ding, P., Huang, J., & Hu, X. (2019). Robust negative sampling for network embedding..

Bavelas, A. (1950). Communication patterns in task-oriented groups. *The Journal of the Acoustical Society of America*, *22*(6), 725–730.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., & Weinberger, K. Q. (Eds.), *Advances in Neural Information Processing Systems 26*, pp. 2787–2795. Curran Associates, Inc.

Brandes, U. (2008). On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, *30*(2), 136–145.

Chollet, F., et al. (2015). Keras. https://keras.io.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, *9*, 1871–1874.

Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, *40*(1), 35–41.

Giles, C. L., Bollacker, K. D., & Lawrence, S. (1998). Citeseer: an automatic citation indexing system. In *INTERNATIONAL CONFERENCE ON DIGITAL LIBRARIES*, pp. 89–98. ACM Press.

Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks.. cite arxiv:1607.00653 Comment: In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

Kotnis, B., & Nastase, V. (2017). Analysis of the impact of negative sampling on link prediction in knowledge graphs. *CoRR*, *abs/1708.06816*.

Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.*, *58*(7), 1019–1031.

McCallum, A. (2017). Cora dataset..

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, *abs/1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., & Weinberger, K. Q. (Eds.), *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013c). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119.

Mikolov, T., Yih, S. W.-t., & Zweig, G. (2013d). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*. Association for Computational Linguistics.

Neville, J., & Jensen, D. (2000). Iterative classification in relational data. In *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, pp. 13–20.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pp. 161–172, Brisbane, Australia.

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 701–710, New York, NY, USA. ACM.

Pimentel, T., Veloso, A., & Ziviani, N. (2018). Fast node embeddings: Learning ego-centric representations..

Recht, B., Re, C., Wright, S., & Niu, F. (2011). Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., & Weinberger, K. Q. (Eds.), *Advances in Neural Information Processing Systems 24*, pp. 693–701. Curran Associates, Inc.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015). Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pp. 1067–1077, New York, NY, USA. ACM.

Wang, X., Tang, L., Gao, H., & Liu, H. (2010). Discovering overlapping groups in social media. In *the 10th IEEE International Conference on Data Mining series (ICDM2010)*, Sydney, Australia.

Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. In *AAAI*.

Zhang, D., Yin, J., Zhu, X., & Zhang, C. (2018). Network representation learning: A survey. *IEEE transactions on Big Data*.