Mineral: Multi-modal Network Representation Learning

Zekarias T. Kefato*, Nasrullah Sheikh*, and Alberto Montresor

University of Trento, Trento, Italy
{zekarias.kefato,nasrullah.sheikh,alberto.montresor}@unitn.it

Abstract. Network representation learning (NRL) is a task of learning an embedding of nodes in a low-dimensional space. Recent advances in this area have achieved interesting results; however, as there is no solution that fits all kind of networks, NRL algorithms need to be specialized to preserve specific aspects of the networks, such as topology, information content, and community structure. One aspect that has been neglected so far is how a network reacts to the diffusion of information. This aspect is particularly relevant in the context of social networks. Studies have found out that diffusion reveals complex patterns in the network structure that are otherwise difficult to be discovered by other means. In this work, we describe a novel algorithm that combines topology, information content and diffusion process, and jointly learns a high quality embedding of nodes. We performed several experiments using multiple datasets and demonstrate that our algorithm performs significantly better in many network analysis tasks over existing studies.

Keywords: NRL, Diffusion patterns, Cascades

1 Introduction

Network representation learning (NRL) is the task to embed nodes of a network into a low-dimensional space, while preserving important aspects of the original network. This strategy is an invaluable tool to tackle a variety of subsequent network analysis problems, such as node classification, link prediction, and visualization. It is not only a hard and daunting task to manually engineer high-quality features for the aforementioned problems, but also the resulting features lack the capability of being applicable across different problems. For example, features that are engineered for node classification might not be suitable for link prediction or vice versa; therefore, one has to develop a new set of features for almost every new task.

Automatic network embedding approaches [1,2,3,4,5,6,7,8,9], however, are highly effective in capturing interesting patterns that are applicable to a range of tasks. They are well-suited for learning features that are otherwise difficult to find even for experts. Such techniques have been employed in multiple disciplines, such as speech recognition and signal processing [10] and object recognition [11,12], improving previous state-of-the-art solutions by several orders of magnitude [13].

* Both authors contributed equally to this work.

Recent studies in representation learning through neural networks have achieved remarkable results [10,11,12]. An interesting aspect that makes these model attractive is that different components of the model, called neurons, are activated while detecting different kinds of patterns. In other words, the learned embedding has a set of discriminative features that are shared among different tasks [13]. This is one of the main reasons that made the representations learned using this technique applicable across multiple tasks [13].

There have been a plethora of studies [1,2,3,4,5,6,7,8] that apply neural networks to NRL. The goal of such studies is usually to learn a representation that preserves one or more of the following properties of nodes: (i) neighborhood structure, (ii) content/attribute information, (iii) community affiliation.

First of all, a high-quality embedding should enable to effectively reconstruct the original network. Therefore, preserving the structural information is of paramount importance. A second aspect to be considered is that approaches that incorporate content/attribute information and enforce a constraint on an embedding algorithm to preserve it, achieve higher-quality embeddings compared to content-oblivious approaches [4,6], sometimes by over an order of magnitude.

While significant improvements over traditional techniques have been obtained, there are still several aspects of information networks that reveal interesting properties of the network. For example, it has been observed that the dynamics of diffusion of influence and information (cascades) unveil complex patterns of the network that are effective in identifying groups of users [14,15].

To complement existing studies of NRL, in this study we propose a novel algorithm that learns an embedding of the network that preserves the topology, the content information, as well as the dynamics of diffusion cascades.

Our approach integrates content and diffusion information into the network structure, without requiring any additional data structure. Based on this, we propose a novel algorithm called MINERAL (Multi-modal Network Representation Learning).

Given that in some datasets, only a fraction of nodes are included in cascades, while in other datasets cascade information is completely missing, we simulate a diffusion process that enables to capture complex local and global network structures. Then, we acquire context information of nodes related to their local neighborhood (directly connected neighbors) and global neighborhood (community membership).

Our contribution can be summarized as follows:

- we combine different aspects of a network that enable learning an effective network embedding;
- we propose a novel scalable algorithm for NRL
- we perform several experiments using multiple datasets and across multiple network analysis tasks.

The rest of the paper is organized as follows. Section 2 introduces the basic concepts and notations and presents the problem statement. Section 3 discusses the proposed algorithm. Section 4 reports the experiments and results. Finally, Section 5 discusses related works; the paper is concluded in Section 6.

2 Preliminary

We start by providing definitions of the data models and describe our problem.

Definition 1. *We consider a network $G = (V, E)$, where V is a set containing n nodes and E is a set containing m edges.*

As in social networks, we assume that the nodes are involved in two types of activities: (i) generating their own content (e.g. posts) and (ii) consuming/spreading others' content. Given a node u, $A(u)$ contains all the pieces of content generated or consumed by u. Content is assumed to be textual; in case of multimedial information, metadata and tags could be used instead. One way to incorporate content information is to add a separate node for each piece of content. However, given that often the goal of incorporating content is to better identify similarities between nodes in the representation learning process, we simply introduce a similarity function π on the edges that is defined as follows.

Definition 2. *We consider a similarity function $\pi : E \to [0, 1]$, such that for any $(u, v) \in E$, $\pi(u, v)$ is equal to the Jaccard similarity between u and v:*

$$\pi(u, v) = \frac{|A(u) \cap A(v)|}{|A(u) \cup A(v)|}$$

If the content is textual, one can easily compute π. For example, consider a user u that actively tweets about *politics* and *religion* and a user v tweeting about *sport* and *politics*. One can construct $A(u)$ and $A(v)$ from the set of keywords extracted from their posts and estimate π. This modeling is simple and efficient, as it requires no additional structure with respect to the existing network; it only associates weights to edges. Unless there is a particular benefit one can gain from adding independent nodes for content, which could be expensive, we argue that such modeling is sufficient.

The final piece of our data model is a set of finite cascades $\mathcal{C}$:

Definition 3. *We consider a set of cascades $\mathcal{C} = \{C_1, \ldots, C_c\}$ of size c, where a cascade $C = [u_1, u_2, \ldots, u_{|C|}]$ is a sequence of finite events, each of them representing the infection of a user by a given contagion.*

We use $C(i) = u_i$ to denote the i-th node of the cascade C. We say that a node u is infected before node v in a cascade C, and we write $u \prec_C v$, if and only if $u = C(i)$, $v = C(j)$ and $i < j$. Given a node u and a context size s, we define the *left-hand side infection context* $C(u; s)^{\prec}$ and the *right-hand side infection context* $C(u; s)^{\succ}$:

$$C(u; s)^{\prec} = \{v : v = C(i) \wedge u = C(j) \wedge j - s \leq i \leq j - 1\}$$
$$C(u; s)^{\succ} = \{v : v = C(i) \wedge u = C(j) \wedge j + 1 \leq i \leq j + s\}$$

Definitions 1–3 represents the input of our problem:

Problem 1. Given a network G, a set of cascades $\mathcal{C}$, a similarity function π, and a dimensional number d, we seek to learn a representation of the network specified by $\Phi : V \to \mathbb{R}^d$, provided that Φ preserves as much as possible (i) the network structure, (ii) the similarity between nodes and (iii) the node infection context.

3 Mineral

In this section, we present a detailed description of MINERAL, which exploits two sources of information: in SPC-Mode (Structure+Content-Mode), it uses structural information (the network G) as well as content information associated to nodes (the function π). In CSD-Mode (Cascade-Mode), it utilizes the observed diffusion information (the set of cascades C).

Thanks to function π, the network G can be considered as a weighted graph. Hence without requiring additional structures, we can design an effective algorithm to learn the representation of the network that preserves both structural and content similarity between nodes. One strategy that has proved to be effective for NRL is to use a similar approach to word representation learning in natural language documents. In word representation, the basic idea is to learn a representation of words by predicting their context. Nonetheless, unlike words in a document where their context is obvious as a result of their linear structure, we do not have a straightforward way to deduce the context of nodes in a network. Several strategies have been developed in the literature to address this problem.

In this work, we extend existing approaches based on random walks [1,2] by considering instead a diffusion process. It has been observed that the dynamics of diffusion processes reveal complex local and global structural patterns of the network. Therefore we simulate the diffusion of influence or information using the independent cascade (IC) model [16] to obtain context information for nodes. The cascades generated by simulating IC are merged with actual (observed) cascades, when available.

Algorithm 1 shows the high-level steps required to generate cascades. For each node $u \in V$, r cascades are generated starting from u, based on the IC model and using the content similarity π as an unnormalized probability of infection.

When SIMULATEDIFFUSION(G, π, u, h) is invoked, a cascade of size h is generated starting from u. Let $\mathcal{I}_t$ denote the set of nodes infected at time t; the diffusion process works as follows:

1. At time $t = 0$, a cascade sequence is initiated by infecting the current root, *i.e.* $C = [u]$, i.e., $\mathcal{I}_0 = \{u\}$.
2. At time $t > 0$, each node $v \in \mathcal{I}_{t-1}$ makes a single attempt to infect each of its outgoing neighbor $w \in out(v)$ that is not already infected (i.e., $w \notin C$). The infection succeeds with a probability proportional to $\pi(v, w)$; in such case, w is appended to C and it is included in $\mathcal{I}_t$
3. Repeat the process starting from step 2 while $|C| \leq h$

We restrict the size of cascades (the number of infected nodes) to be at most h nodes, because large, viral cascades (unlike non-viral ones) usually do not capture any relevant local or global structural relation of nodes [14,15].

Generated cascades, together with existing ones if available, are thus used to learn embeddings. Since cascades are sequences of nodes, we borrow the SKIPGRAM [17] model for word representation learning to perform network representation learning. For the purpose of being self-contained, we briefly describe the SKIPGRAM [17] model in our context.

CascadeGenerator(G, π, r, h)

$\mathcal{C} = \emptyset$
for $u \in V$ **do**
  repeat r **times**
    $C =$ simulateDiffusion (G, π, u, h)
    $\mathcal{C}$.insert(C)
return $\mathcal{C}$

SkipGram Given a center node $u \in C$, this model maximizes the log probability of observing context nodes $v \in C(u;s)^{\preceq}$ and $w \in C(u;s)^{\succeq}$ within a window size s. Based on the assumption that the likelihood of observing each context node given a center node is independent, more formally the SkipGram model optimizes the objective in Eq. 1 with respect to the model parameter Φ.

$$\max_{\Phi} \sum_{u \in V} \log Pr(C(u;s)^{\preceq} \mid \Phi(u)) + \log Pr(C(u;s)^{\succeq} \mid \Phi(u)) \tag{1}$$

$$\log Pr(C(u;s)^{D} \mid \Phi(u)) = \sum_{v \in C(u;s)^{D}} \log Pr(v|\Phi(u)) \tag{2}$$

where D is either $\preceq$ or $\succeq$, and $\Phi(u) \in [0,1]^d$ is a $d-$dimensional representation of u. The right-hand side term in Eq. 2 is specified using the softmax function:

$$Pr(v|\Phi(u)) = \frac{\exp(\Phi(v)^T \cdot \Phi(u))}{\sum_{w \in N} \exp(\Phi(w)^T \cdot \Phi(u))} \tag{3}$$

Nonetheless, directly estimating the conditional probability in Eq. 3 is expensive, because of the normalization constant that needs to be computed for every node. For this reason, different approximation strategies have been suggested in the literature; in this work, we adopt the "Negative Sampling" strategy [17] that characterizes a good model by its power to discriminate appropriate context nodes from noise. Then, the computation of $\log Pr(v|\Phi(u))$ using the negative sampling strategy is shown in Eq. 4.

$$\log Pr(v|\Phi(u)) = \log \sigma(\Phi(v)^T \Phi(u)) + neg(u;l) \tag{4}$$

σ is the logistic function, and we need our model to effectively differentiate v from the l negative examples drawn from some noise distribution $\mathcal{N}(u)$ of u, where $neg(u;l)$ is the noise model and is defined as:

$$neg(u;l) = \sum_{i=1}^{l} \mathbf{E}_{w_i \sim \mathcal{N}(u)}[-\log \sigma(\Phi(w_i)^T \Phi(u))] \tag{5}$$

Numerically, a good model should produce a small expected probability for the noise model and larger probability for the data model (the first term on the right-hand-side of Eq. 4).

Finally, we employ the stochastic gradient descent algorithm to minimize the negative log-likelihood of the objective in Eq. 1 based on the negative sampling strategy in Eq. 4, 5 and obtain the complete model parameters $\Phi \in V \rightarrow [0,1]^d$.

Dataset	$\|\mathbf{V}\|$	$\|\mathbf{E}\|$	$\|\mathcal{C}\|$	Number of labels	Type of labels
Twitter	595,460	14,273,311	397,681	5	top-5 communities
Memetracker	3,836,314	15,540,787	71,568	5	top-5 communities
Flickr	80,513	5,899,882	-	195	Groups
Blogcatalog	10,312	333,983	-	39	Interests

Table 1. Summary of the datasets

4 Experiments and Results

In order to demonstrate the effectiveness of our algorithm, we have carried out several experiments across multiple network analysis problems using multiple datasets, listed below. A brief summary of the characteristics of the datasets is given in Table 1.

- Twitter [14]: a dataset containing the follower network of Twitter users and cascade information of hashtags. Each time a user adopts a hashtag (by creating a new or using an existing one), it is added to the set of her keywords. A cascade is constructed by sorting the users according to their first use of a particular hashtag.
- Memetracker [18]: a dataset containing the interaction history between different news media and blog web pages during a year. Each page is associated with a set of memes, which are considered as its keywords. Memes are grouped into clusters, and we consider each cluster id as a contagion that has infected every page that has mentioned a meme that belongs to the cluster. Similar to Twitter, cascades are built by sorting the users of a contagion according to the time of first use.
- Blogcatalog [19]: a dataset containing a network of bloggers. There are 39 topic categories which are considered as content information for each author.
- Flickr [19]: a photo sharing site paired to a social network. Users place their pictures under a set of predefined categories which can be considered content information.

For Twitter and Memetracker, users are labeled based on their communities. First we identify the (non-overlapping) community to which a user belongs using [20], and then we associate it as her label. We utilize both SPC and CSD modes for these datasets, since information regarding structure, content, and cascades is available. In addition, in all the experiments we have used $h = 500$ for Twitter and Memetracker, $h = 200$ for Flickr and $h = 50$ for Blogcatalog.

4.1 Baselines

Existing methods [6,4] that consider content information are usually based on matrix factorization, which makes them unscalable for large networks. For this reason, we only consider the following two content-oblivious approaches as baseline methods:

Algorithm	P@100	P@500	P@1000	P@5000	P@10000	p@50000	p@100000	p@500000
MINERAL	**99.9**	99.8	99.8	99.8	**99.8**	**99.8**	**99.8**	**99.0**
DEEPWALK	96.6	97.0	97.1	97.1	97.1	97.1	97.1	96.9
Line	99.3	99.8	**99.9**	99.8	99.7	98.5	94.5	71.0

Table 2. Result for the link prediction task on the Twitter dataset

Algorithm	P@100	P@500	P@1000	P@5000	P@10000	p@50000	p@100000	p@500000
MINERAL	**100.0**	**99.9**	**99.9**	**99.6**	**99.5**	**99.5**	**99.4**	98.6
DEEPWALK	99.1	99.0	99.0	99.1	99.0	99.0	99.0	**99.0**
Line	91.2	92.2	89.9	85.2	83.3	72.8	68.9	65.4

Table 3. Result for the link prediction task on the Memetracker dataset

Algorithm	P@50	P@100	P@500	P@1000	p@5000	p@10000	p@50000	p@100000
MINERAL	**99.2**	**99.6**	**99.6**	**99.6**	**99.4**	**99.2**	97.4	94.9
DEEPWALK	96.6	96.6	97.4	97.5	97.5	97.5	97.4	**97.1**
LINE	54.4	61.0	61.6	58.8	51.6	48.9	44.2	42.5

Table 4. Result for the link prediction task on the Flickr dataset

1. DEEPWALK [1]: is a method that utilizes truncated random walks for network embedding, where each step of a walk is chosen uniformly at random. Equivalent to the current work, they use the SKIPGRAM model and it is trained using the walks.
2. LINE [3]: is a proximity based approach, trained by concatenating two independently trained models based on the notions of first-order and second-order similarity of nodes. In other words, in the first phase they train a model that preserves the undirected link structure between nodes; in the second phase, they train a model that preserves the directed or undirected 2-hop link structure of the network.

4.2 Link Prediction

Link prediction is one of the most important network analysis problems. There are three main techniques solving it, based on node similarity, topology, and social theory [21]. Very often, such techniques rely on experts to craft informative features that enable to effectively predict links, and this makes them expensive. Instead of manually-crafted features, we use here the learned embeddings to perform link prediction. Towards this end, we randomly sampled 15% of the existing edges from the network; we also randomly sampled the same amount of node pairs that are not in the edge set. We then used the learned embedding to effectively predict the links. That is, given a pair of nodes $\{u, v\} \subseteq V$, we compute the probability $p(u, v)$ of an edge existing between the two nodes as:

$$p(u, v) = \frac{1}{1 + e^{-(\Phi(u)^T \cdot \Phi(v))}}$$

	Training Ratio								
Algorithm	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
MINERAL	**98.19**	**98.05**	**97.97**	**97.98**	**97.95**	**97.91**	**97.74**	**97.51**	**96.93**
DEEPWALK	97.78	97.76	97.86	97.67	97.61	97.45	97.42	97.02	96.01
LINE	84.19	85.74	85.02	85.11	85.18	84.69	84.06	82.20	76.19

Table 5. Node classification accuracy on different levels of labeled training set ratio for the Twitter dataset

	Training Ratio								
Algorithm	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
MINERAL	**98.19**	**98.05**	**97.97**	**97.98**	**97.95**	**97.91**	**97.74**	**97.51**	**96.93**
DEEPWALK	97.78	97.76	97.86	97.67	97.61	97.45	97.42	97.02	96.01
LINE	84.19	85.74	85.02	85.11	85.18	84.69	84.06	82.20	76.19

Table 6. Node classification accuracy on different levels of labeled training set ratio for the Memetracker dataset

Then we sort the predicted edges according to $p(u, v)$ in descending order and evaluate the performance of an embedding in correctly predicting the edges using the precision-at-K (P@K) score. P@K measures the fraction of correctly predicted edges on the top-K results, i.e what percent of the top-K edges are true edges from the randomly sampled edges. For each K value we perform the experiments 10 times and report the average. Tables 2, 3 and 4 show the results for the Twitter, Memetracker and Flickr datasets; MINERAL performance is as good as or better than the baselines.

4.3 Node Label Classification

The second problem we addressed is label classification. We consider two instance of it, namely multi-class and multi-label classifications. For the Twitter and Memetracker datasets, we tackled the multi-class classification problem, because–as shown in Table 1–labels are communities and each node belongs to just a single community. In the other datasets, given that multiple labels are present, we performed multi-label classification. To evaluate the effectiveness of a model in the classification task, we adopt the same evaluation metrics as in previous studies, and hence we use Accuracy, F1-Micro and F1-Macro metrics.

The Multi-class classification results for the Twitter and Memetracker datasets are reported in Table 5 and 6, respectively. Similar to previous studies, we performed these experiments on different fractions of labeled training sets (Training Ratio $\in \{10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%, 90\%\}$). Under this setting, accuracy is the evaluation metric; and as shown in the tables, MINERAL performs slightly better than DEEPWALK and significantly better than LINE. For the other datasets, however, MINERAL significantly outperforms both baselines in multi-label classification. Figure 1 and 2 report the results on different training ratios (x-axis) using F1-Micro and F1-Macro measures (y-axis).

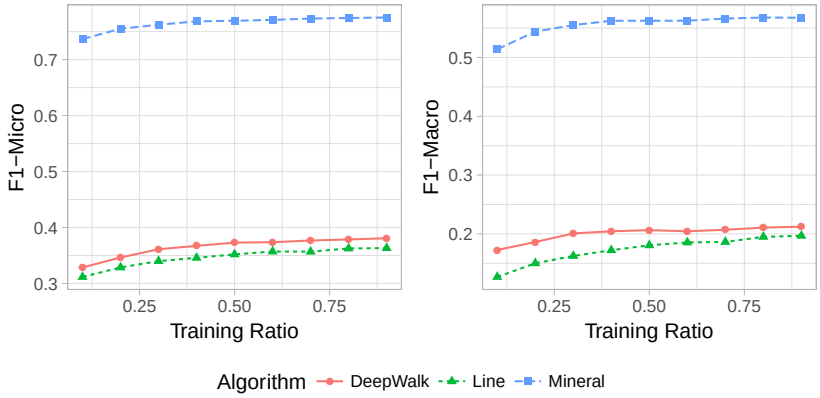

Fig. 1. Multi-label classification (using one-vs-rest logistic regression classifier) on the Blogcatalog dataset

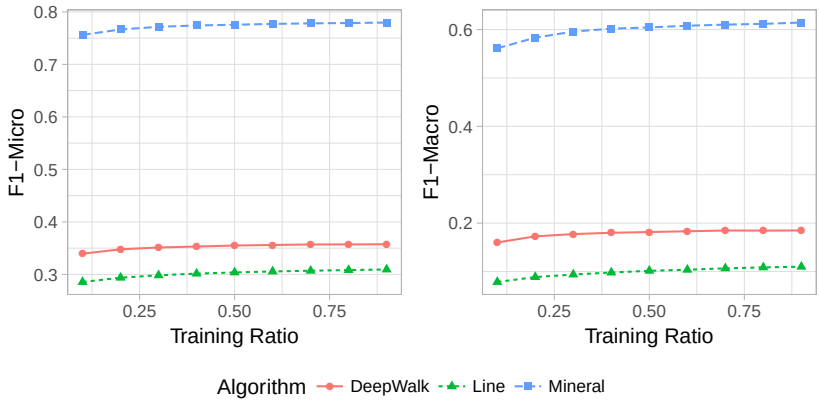

Fig. 2. Multi-label classification (using one-vs-rest logistic regression classifier) on the Flickr dataset

4.4 Network Visualization

The last but not the least application of NRL is network visualization. We use the Twitter dataset for this task, and the visualization is performed using t-Distributed Stochastic Neighbor Embedding (t-SNE) [22]. Given a set of q communities, an informative visualization should maintain a knit cluster for members of the same community and maintain clear boundaries between different communities. As shown in Fig. 3, MINERAL's visualization gives the best result. Members of each community are densely clustered and are far from members of other communities.

4.5 Parameter Sensitivity

Now we turn into analyzing the sensitivity of the virality controlling parameter, which is h. In Section 3 we have argued that "viral" or large cascades do not

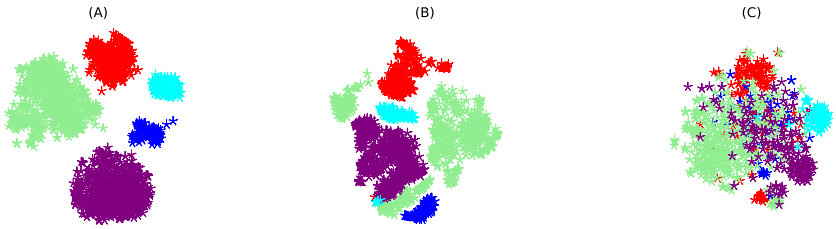

Fig. 3. Visualization of top-5 communities with atmost 2000 users in the Twitter Dataset using (A) MINERAL (B) DEEPWALK and (C) Line

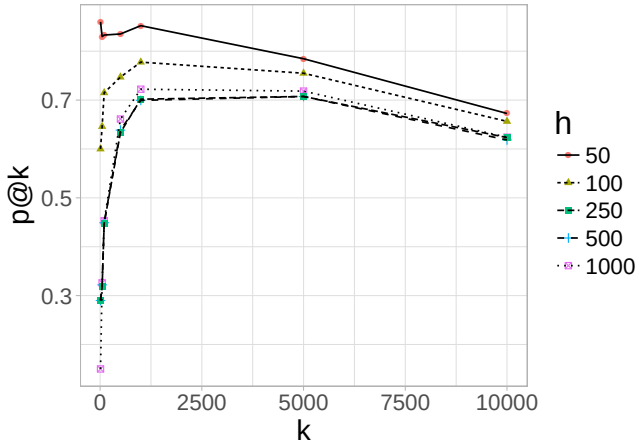

Fig. 4. Sensitivity of the parameter h using the link prediction task on Blogcatalog

capture any meaningful dependency between infected nodes of the cascades. To empirically prove that such is the case, we have performed experiments over different values of $h \in \{50, 100, 250, 500, 1000\}$ on the Blogcatalog dataset. As shown in Fig 4, the precision@k significantly drops as we increase the size of h. For example, for a fixed $k = 10$, the precision@k is $P@k = 0.86$ for $h = 50$, $P@k = 0.6$ for $h = 100$, $P@k = 0.29$ for $h = 500$, and $P@k = 0.15$ for $h = 1000$.

5 Related Work

Recent advances in neural network models have attracted researches from several communities such as computer vision, NLP, and social network analysis. In the last two communities in particular, a seminal work of Mikolov et al. [17] in representation learning (embedding) of words in documents using a shallow neural network model has inspired studies [1,2] in network representation learning. Among the approaches introduced for word embedding, the Skip-Gram model [17] is the one that has been most largely used for network representation learning. The Skip-Gram model is used to learn a representation of words by way

of predicting context words. The context of a node in a network, however, does not have a straightforward definition. Studies have introduced different strategies of capturing nodes context, for example using random walks [1,2], pair-wise proximities [3,5], and community structures [8,7]. Once a context is formalized, different neural network (based on either shallow or deep models) are employed for the representation learning task. Then the learned representations are utilized for downstream network analysis tasks.

Studies such as [6] propose a NRL algorithm based on matrix factorization. Such techniques, however, are computationally expensive and not scalable for large networks.

6 Conclusion

This study presents MINERAL, a novel algorithm for network representation learning (NRL) that leverages three network aspects: topology, node content, and diffusion. The algorithm efficiently encodes content information associated with nodes into a similarity function between pairs of connected nodes. Then it combines the network and similarity information with natural (observed) or simulated cascades, and acquires context information of nodes. Finally, we combine everything as a set of cascades and employ the SKIPGRAM model to learn an embedding that preserves structural, content, and diffusion context of nodes.

We performed several experiments using multiple datasets across several network analysis problems, and compared the performance of our approach with existing NRL baseline methods. Our results show that MINERAL significantly outperforms the baselines specially in multi-label classification and network visualization. It also performs slightly better than the baselines in link prediction. Even though our data modeling is effective in capturing many kinds of content information, in this study we have focused on textual information.

Acknowledgements This research was partially supported by EIT Digital Project Sensemaking Service: Entity Linking for Big Linked Data - Act. #17151 - 2017.

References

1. B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. of the 20th ACM Int. Conf. on Knowledge Discovery and Data Mining*, KDD '14, pp. 701–710, ACM, 2014.
2. A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. of the 22Nd ACM Int. Conf. on Knowledge Discovery and Data Mining*, KDD '16, pp. 855–864, ACM, 2016.
3. J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: large-scale information network embedding," *CoRR*, vol. abs/1503.03578, 2015.
4. X. Huang, J. Li, and X. Hu, "Label informed attributed network embedding," in *Proc. of the Tenth ACM Int. Conf. on Web Search and Data Mining*, WSDM '17, pp. 731–739, ACM, 2017.

5. D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. of the 22Nd ACM Int. Conf. on Knowledge Discovery and Data Mining*, KDD '16, pp. 1225–1234, ACM, 2016.
6. C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information," in *Proc. of the 24th Int. Conf. on Artificial Intelligence*, IJCAI'15, pp. 2111–2117, AAAI Press, 2015.
7. C. Tu, H. Wang, X. Zeng, Z. Liu, and M. Sun, "Community-enhanced network representation learning for network analysis," *CoRR*, vol. abs/1611.06645, 2016.
8. X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, "Community preserving network embedding," *AAAI*, 2017.
9. R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation learning of knowledge graphs with entity descriptions," in *Proc. of the Thirtieth AAAI Conf. on Artificial Intelligence*, AAAI'16, pp. 2659–2665, AAAI Press, 2016.
10. G. E. Dahl, M. Ranzato, A.-r. Mohamed, and G. Hinton, "Phone recognition with the mean-covariance restricted boltzmann machine," in *Proc. of the 23rd Int. Conf. on Neural Information Processing Systems*, NIPS'10, pp. 469–477, Curran Associates Inc., 2010.
11. D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *2012 IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3642–3649, June 2012.
12. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, June 2015.
13. Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 1798–1828, Aug 2013.
14. L. Weng, F. Menczer, and Y.-Y. Ahn, "Virality prediction and community structure in social networks," *Sci. Rep.*, vol. 3, no. 2522, 2013.
15. L. Weng, F. Menczer, and Y.-Y. Ahn, "Predicting successful memes using network and community structure.," in *ICWSM* (E. Adar, P. Resnick, M. D. Choudhury, B. Hogan, and A. H. Oh, eds.), The AAAI Press, 2014.
16. D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proc. of the Ninth ACM Int. Conf. on Knowledge Discovery and Data Mining*, KDD '03, pp. 137–146, ACM, 2003.
17. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. of the 26th Int. Conf. on Neural Information Processing Systems*, NIPS'13, pp. 3111–3119, Curran Associates Inc., 2013.
18. J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *Proc. of the 15th ACM Int. Conf. on Knowledge Discovery and Data Mining*, KDD '09, pp. 497–506, ACM, 2009.
19. L. Tang and H. Liu, "Relational learning via latent social dimensions," in *Proc. of the 15th ACM Int. Conf. on Knowledge Discovery and Data Mining*, KDD '09, pp. 817–826, ACM, 2009.
20. V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.
21. P. Wang, B. Xu, Y. Wu, and X. Zhou, "Link prediction in social networks: the state-of-the-art," *Sci. China Inform. Sci.*, vol. 58, no. 1, pp. 1–38, 2015.
22. L. van der Maaten and G. Hinton, "Visualizing high-dimensional data using t-sne," *Journal of Machine Learning Research*, 2008.