Max-Margin DeepWalk: Discriminative Learning of Network Representation

Cunchao Tu[1,2*], **Weicheng Zhang**[3*], **Zhiyuan Liu**[1,2†], **Maosong Sun**[1,2]
[1]Department of Computer Science and Technology,
State Key Lab on Intelligent Technology and Systems,
National Lab for Information Science and Technology,
Tsinghua University, Beijing, China
[2]Jiangsu Collaborative Innovation Center for Language Ability,
Jiangsu Normal University, Xuzhou, China
[3]Beijing University of Posts and Telecommunications, China

Abstract

DeepWalk is a typical representation learning method that learns low-dimensional representations for vertices in social networks. Similar to other network representation learning (NRL) models, it encodes the network structure into vertex representations and is learnt in unsupervised form. However, the learnt representations usually lack the ability of discrimination when applied to machine learning tasks, such as vertex classification. In this paper, we overcome this challenge by proposing a novel semi-supervised model, max-margin DeepWalk (MMDW). MMDW is a unified NRL framework that jointly optimizes the max-margin classifier and the aimed social representation learning model. Influenced by the max-margin classifier, the learnt representations not only contain the network structure, but also have the characteristic of discrimination. The visualizations of learnt representations indicate that our model is more discriminative than unsupervised ones, and the experimental results on vertex classification demonstrate that our method achieves a significant improvement than other state-of-the-art methods. The source code can be obtained from https://github.com/thunlp/MMDW.

1 1 Introduction

Network representation plays a critical role in the area of network analysis. An effective network representation is helpful to many network analysis tasks, such as vertex classification, clustering and link prediction. As a basic component in network, every vertex is typically represented as a discrete symbol, which is called one-hot representation. Due to its simplicity, such representation method has been widely adopted for network analysis. However, one-hot representation usually suffers from the sparsity issue and does not fully consider relatedness between vertices.

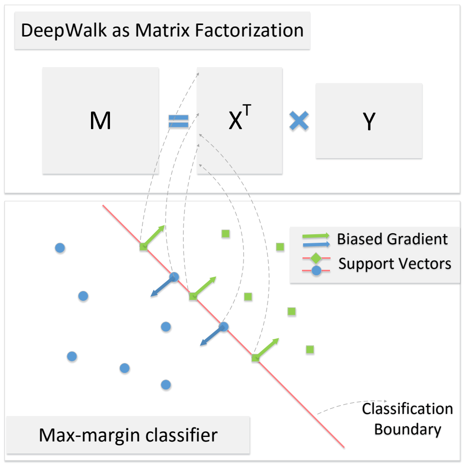


Figure 1: An illustration of Max-Margin DeepWalk.

Motivated by the ideology of distributed representation learning in recent years, network representation learning (NRL) is proposed to address these issues. NRL aims to learn a real-valued vector for each vertex to reflect its network information. The learnt vectors are capable of various network analysis tasks, such as vertex classification and link prediction. The relatedness between vertices can also be achieved on the basis of real-valued vectors.

There have been a number of NRL models proposed in recent years, such as DeepWalk [Perozzi *et al.*, 2014] and LINE [Tang *et al.*, 2015]. [Perozzi *et al.*, 2014] proposed an online NRL model, DeepWalk, that learns vertex representations based on local network information. It conducts random walks to obtain vertex sequences. With a mass of sequences, DeepWalk employs Skip-Gram [Mikolov *et al.*, 2013a], an efficient word representation learning model, to learn vertex representations by treating vertex sequences as word sentences. This straightforward analogy between words

*Indicates equal contribution
†Corresponding Author: Zhiyuan Liu (liuzy@tsinghua.edu.cn)

and vertices has been verified on multi-label network classification tasks.

Most of the previous NRL models are learnt in unsupervised schemas. Though the learnt representations can be applied to various tasks, they can be weak in the particular prediction task. It's worth pointing out that there are many additional labeling information for social network vertices in real world. For example, pages in Wikipedia[1] can have their categorical labels, like "arts", "history", "science"; papers in Cora and Citeseer are also stored with field labels for easy retrieval. Such labeling information usually contains a useful summarization or features of the vertices, but is not directly utilized in the original network representation learning models.

It is nontrivial to explore how to integrate labeling information into network representation learning and learn discriminative representations for network vertices. Inspired by max-margin principle, we propose max-margin DeepWalk (MMDW), a discriminative NRL model, to seek predictive representations for vertices in social networks.

As illustrated in Fig. 1, MMDW firstly learns DeepWalk as matrix factorization. Afterwards, it trains a max-margin based classifier (e.g., support vector machine [Hearst *et al.*, 1998]) and enlarges the distance between support vectors and classification boundaries. In other words, MMDW jointly optimizes the max-margin based classifier (e.g., support vector machine) and NRL model. Influenced by max-margin classifier, the representations of vertices are therefore more discriminative and more suitable for prediction tasks, which we will demonstrate in the following sections.

In conclusion, we make several noteworthy contributions as follows:

(1) We propose a discriminative NRL model, max-margin DeepWalk, to incorporate labeling information into vertex representations. As a semi-supervised model, it applies max-margin principle to network representation learning for the first time.

(2) We put forward the idea of **Biased Gradient** in NRL. The biased gradient of a vector indicates the direction where the vector should move towards. The movement is able to enlarge the margin between two categories, and is conducted in the gradient form in gradient descent algorithms.

(3) We conduct vertex classification experiments on several real-world datasets to verify the effectiveness of MMDW. The experimental results demonstrate that MMDW significantly outperforms traditional NRL models. It achieves 5% to 10% improvements compared to typical NRL methods. Besides, we also compare the visualizations using t-SNE to illustrate the discrimination of MMDW.

2 2 Related Work

Network representation learning (NRL) aims to encode network structure into a low-dimensional space. The learnt representations can be adopted to many network analysis tasks, such as vertex classification, clustering and link prediction.

Many NRL models have been proposed to learn efficient vertex representations. Inspired by Skip-Gram [Mikolov *et al.*, 2013a], a widely adopted word representation learning model in NLP, [Perozzi *et al.*, 2014] proposed DeepWalk by corresponding vertices to words. DeepWalk performs random walks to generate vertex sequences and trains Skip-Gram model to obtain vertex representations. Derived from Skip-Gram, DeepWalk has been extensively verified on various network analysis tasks. Another typical NRL model is LINE [Tang *et al.*, 2015], which is proposed to handle large-scale networks with millions of vertices and billions of edges.

There have been some max-margin based learning methods in other fields. [Roller, 2004] firstly introduced max-margin principle into markov networks. [Zhu *et al.*, 2012] proposed maximum entropy discrimination LDA (MedLDA) to learn a discriminative topic model (e.g., latent Dirichlet allocation [Blei *et al.*, 2003]). Besides, max-margin also benefits many NLP tasks, such as semantic parsing [Taskar *et al.*, 2004] and word segmentation [Pei *et al.*, 2014].

However, to the best of our knowledge, there is little work learning discriminative network representations with labeling information considered. Most of the mentioned NRL methods are learnt in unsupervised fashions. To fill this gap, we propose max-margin DeepWalk (MMDW) to learn discriminative representations for vertices in social networks.

3 3 The Framework

In this section, we present a novel semi-supervised social representation model, max-margin DeepWalk (MMDW), that utilizes the labeling information when learning vertex representations. MMDW is a unified learning framework based on matrix factorization. In this model, we optimize the max-margin based classifier (SVM) as well as the aimed matrix factorization model. In contrast, the conventional approaches usually learn social representations without leveraging the labeling information and apply the learnt representations into classification tasks. The vertices' labels are not able to influence the way representations are learnt. Therefore, the learnt representations are often found not so discriminative.

3.1 3.1 Formalizations

Let us begin with formally defining the problem of social representation learning. Suppose there is a social network $G = (V, E)$, where V is the set of all vertices, and E are connections between these vertices, i.e., $E \subset V \times V$, social representation learning aims to build a low-dimensional representation $\mathbf{x}_v \in \mathbb{R}^k$ for each vertex $v \in V$, where k is the the dimension of representation space and expected much smaller than $|V|$. The learnt representations encode semantic roles of vertices in the social network, which can be used to measure relatedness between vertices, and can also play as features for classification tasks. With the corresponding label $l \in \{1, \cdots, m\}$, classifiers like logistic regression and SVM can be trained.

In the following parts, we introduce a typical social representation model, DeepWalk, and its matrix factorization form. Afterwards, we give detailed introduction to max-margin DeepWalk.

3.2 3.2 DeepWalk as Matrix Factorization

DeepWalk [Perozzi *et al.*, 2014] performs random walks over a network to build vertex sequences. By regarding vertex se-

[1]https://en.wikipedia.org

quences as word sequences, it adopts Skip-Gram [Mikolov *et al.*, 2013b], a widely-used word representation algorithm, to learn network representations. The performance of DeepWalk has been extensively verified on multiple tasks and datasets.

Motivated by Skip-Gram, DeepWalk aims to maximize the co-occurrence probability between a target vertex and its context vertices within a random-walk window. Formally, suppose we have a random walk sequence $\mathbf{s} = \{v_1, \ldots, v_M\}$ with each $v_i \in V$. We set a window size K, and for each target vertex v_i, we define its context vertices $\mathbf{c}_i = (v_{i-K}, \ldots, v_{i+K}) \setminus v_i$. Thus, the objective of DeepWalk can be formalized as follows,

$$\mathcal{L}(S) = \sum_{\mathbf{s} \in S} [\frac{1}{M} \sum_{i=K}^{M-K} \sum_{v_j \in \mathbf{c}_i} \log \Pr(v_j | v_i)]. \tag{1}$$

Here, S is the set of random walk sequences generated from random walks. The probability $\Pr(v_j|v_i)$ is computed using softmax function,

$$\Pr(v_j|v_i) = \frac{\exp(\mathbf{x}_j \cdot \mathbf{x}_i)}{\sum_{t \in V} \exp(\mathbf{x}_t \cdot \mathbf{x}_i)}, \tag{2}$$

where $\mathbf{x}_j$ and $\mathbf{x}_i$ are the representation vectors of the vertices v_j and v_i, and $(\cdot)$ is the inner product between vectors.

[Yang *et al.*, 2015] proved that DeepWalk actually factorizes a matrix M. Each entry in M is formalized as

$$M_{ij} = \log \frac{[e_i(A + A^2 + \cdots + A^t)]_j}{t}. \tag{3}$$

Here, A is the transition matrix, which can be seen as the row normalized adjacency matrix. e_i denotes an indicator vector, in which the i-th entry is 1 and the others are all 0. The entry v_{ij} is logarithm of the average probability that vertex i walks to vertex j in t steps.

From Eq. 3, we find that computing an precise M is time-consuming. Thus, we employ the settings in [Yang *et al.*, 2015], by factorizing the matrix as $M = (A + A^2)/2$. In practice, we factorize M instead of $\log M$, since $\log M$ has much more non-zero entries than M and the complexity of matrix factorization is proportional to the number of non-zero entries [Yu *et al.*, 2014].

Now, we formalize the DeepWalk using matrix factorization $M = X^T Y$. We aim to find matrices $X \in \mathbb{R}^{k \times |V|}$ and $Y \in \mathbb{R}^{k \times |V|}$ to minimize

$$\min_{X,Y} \mathcal{L}_{DW} = \min_{X,Y} ||M - (X^T Y)||_2^2 + \frac{\lambda}{2}(||X||_2^2 + ||Y||_2^2), \tag{4}$$

where the factor λ controls the weight of regularization part.

3.3 3.3 Max-Margin DeepWalk

Max-margin methods, such as support vector machines (SVMs) [Hearst *et al.*, 1998], are usually applied to various discriminative problems including document categorization and handwritten character recognition.

In this work, we take the learnt representations X as features and train an SVM for vertex classification. Suppose the training set is $\mathcal{T} = \{(\mathbf{x}_1, l_1), \cdots, (\mathbf{x}_T, l_T)\}$, the multi-class SVM aims to find optimal linear functions by solving the following constrained optimization problem:

$$\begin{aligned} \min_{W,\xi} \mathcal{L}_{SVM} &= \min_{W,\xi} \frac{1}{2} \|W\|_2^2 + C \sum_{i=1}^{T} \xi_i \\ \text{s.t.} \quad \mathbf{w}_{l_i}^T \mathbf{x}_i &- \mathbf{w}_j^T \mathbf{x}_i \geq e_i^j - \xi_i, \quad \forall i, j \end{aligned} \tag{5}$$

where

$$e_i^j = \begin{cases} 1, & \text{if} \quad l_i \neq j, \\ 0, & \text{if} \quad l_i = j. \end{cases} \tag{6}$$

Here, $W = [\mathbf{w}_1, \cdots, \mathbf{w}_m]^T$ is the weight matrix of SVM, and $\xi = [\xi_1, \cdots, \xi_T]$ is the slack variable that tolerates errors in the training set.

As mentioned in previous parts, such pipeline method can not influence the way vertex representations are learnt. With learnt representations, SVM only helps to find optimal classifying boundaries. As a result, the representations themselves are not discriminative.

Inspired by max-margin learning on topic model [Zhu *et al.*, 2012], we present max-margin DeepWalk (MMDW) to learn discriminative representations of vertices in social networks. MMDW aims to optimize the max-margin classifier of SVM as well as matrix factorization based DeepWalk. The objective is defined as follows:

$$\begin{aligned} \min_{X,Y,W,\xi} \mathcal{L} &= \min_{X,Y,W,\xi} \mathcal{L}_{DW} + \frac{1}{2} \|W\|_2^2 + C \sum_{i=1}^{T} \xi_i \\ \text{s.t.} \quad \mathbf{w}_{l_i}^T \mathbf{x}_i &- \mathbf{w}_j^T \mathbf{x}_i \geq e_i^j - \xi_i, \quad \forall i, j \end{aligned} \tag{7}$$

4 4 Optimization of MMDW

The parameters in the objective 7 include vertex representation matrix X, context representation matrix Y, weight matrix W and slack variable vector ξ. We employ an effective optimization strategy by optimizing the two parts separately. With the introduction of Biased Gradient, the matrix factorization is significantly affected by trained max-margin classifier.

We perform our optimization algorithm as follows.

4.1 4.1 Optimization over W and ξ:

When $\mathbf{X}$ and $\mathbf{Y}$ are fixed, the primal problem 7 becomes the same as a standard multi-class SVM problem proposed by Crammer and Singer (2000), which has the following dual form:

$$\begin{aligned} \min_{\mathbf{Z}} \frac{1}{2} \|W\|_2^2 &+ \sum_{i=1}^{T} \sum_{j=1}^{m} e_i^j z_i^j \\ \text{s.t.} \quad \sum_{j=1}^{m} z_i^j &= 0, \quad \forall i \\ z_i^j &\leq C_{l_i}^j, \quad \forall i, j \end{aligned} \tag{8}$$

where

$$\mathbf{w}_j = \sum_{i=1}^{l} z_i^j \mathbf{x}_i, \quad \forall j$$

and

$$C_{y_i}^m = \begin{cases} 0, & \text{if} \quad y_i \neq m, \\ C, & \text{if} \quad y_i = m. \end{cases}$$

Here, the lagrangian multiplier α_i^j is replaced by $C_{l_i}^j - z_i^j$ for short.

To solve this dual problem, we utilize a coordinate descent method to decompose Z into blocks $[\mathbf{z}_1, \cdots, \mathbf{z}_T]$, where

$$\mathbf{z}_i = [z_i^1, \cdots, z_i^m]^T, \quad i = 1, \cdots, T$$

An efficient sequential dual method [Keerthi *et al.*, 2008] is applied to solve the sub-problem formed by $\mathbf{z}_i$.

4.2 Optimization over X and Y:

When W and ξ are fixed, the primal problem 7 turns into minimizing the square loss of matrix factorization with additional boundary constraints as follows:

$$\min_{X,Y} \mathcal{L}_{DW}(X, Y; M, \lambda) \tag{9}$$
$$\text{s.t.} \quad \mathbf{w}_{l_i}^T \mathbf{x}_i - \mathbf{w}_j^T \mathbf{x}_i \geq e_i^j - \xi_i, \quad \forall i, j$$

Without the consideration of constraints, we have

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial X} &= \lambda X - Y(M - X^T Y), \\ \frac{\partial \mathcal{L}}{\partial Y} &= \lambda Y - X(M - X^T Y). \end{aligned} \tag{10}$$

$\forall i \in \mathcal{T}, j \in 1, \cdots, m$, if $l_i \neq j$ and $\alpha_i^j \neq 0$, according to KKT conditions, we have

$$\mathbf{w}_{l_i}^T \mathbf{x}_i - \mathbf{w}_j^T \mathbf{x}_i = e_i^j - \xi_i. \tag{11}$$

When the decision boundary is fixed, we want to bias such support vector $\mathbf{x}_i$ towards the direction that favors a more accurate prediction. Thus the biased vector can enlarge the discrimination.

Here we explain how the bias is calculated. Given a vertex $i \in \mathcal{T}$, for the j-th constraint, we add a component $\alpha_i^j(\mathbf{w}_{l_i} - \mathbf{w}_j)^T$ to $\mathbf{x}_i$, then the constraint becomes

$$\begin{aligned} &(\mathbf{w}_{l_i} - \mathbf{w}_j)^T(\mathbf{x}_i + \alpha_i^j(\mathbf{w}_{l_i} - \mathbf{w}_j)) \\ &= (\mathbf{w}_{l_i} - \mathbf{w}_j)^T \mathbf{x}_i + \alpha_i^j ||(\mathbf{w}_{l_i} - \mathbf{w}_j||_2^2 \\ &> e_i^j - \xi_i. \end{aligned} \tag{12}$$

Note that, we utilize the lagrangian multiplier α_i^j to judge whether the vector is on the decision boundary. Only $\mathbf{x}_i$ with $\alpha_i^j \neq 0$ is added a bias based on the j-th constraint.

For a vertex $i \in \mathcal{T}$, the gradient becomes $\frac{\partial \mathcal{L}}{\partial \mathbf{x}_i} + \eta \sum_{j=1}^m \alpha_i^j (\mathbf{w}_{l_i} - \mathbf{w}_j)^T$, which is named Biased Gradient. Here, η balances the primal gradient and the bias.

Before X is updated, W and ξ satisfy the KKT conditions of SVM, and this solution is initially optimal. But after updating X, the KKT conditions do not hold. This will lead to a slight increase of the objective, but this increase is usually within an acceptable range according to our experiments.

5 5 Experiments

In this section, we conduct vertex classification in social networks to evaluate our proposed model. Besides, we also visualize the learnt representations to verify that MMDW is able to learn discriminative representations.

5.1 5.1 Datasets and Experiment Settings

We employ the following three typical datasets for vertex classification.

Cora. Cora[2] is a research paper set constructed by [McCallum *et al.*, 2000]. It contains $2,708$ machine learning papers which are categorized into 7 classes. The citation relationships between them form a typical social network.

Citeseer. Citeseer is another research paper set constructed by [McCallum *et al.*, 2000]. It contains $3,312$ publications and $4,732$ connections between them. These papers are from 6 classes.

Wiki. Wiki [Sen *et al.*, 2008] contains $2,405$ web pages from 19 categories and $17,981$ links between them. It's much denser than Cora and Citeseer.

For evaluation, we randomly sample a portion of labeled vertices and take their representations as features for training, and the rest are used for testing. We increase the training ratio from 10% to 90%, and employ multi-class SVM [Crammer and Singer, 2002] to build classifiers.

5.2 5.2 Baseline Methods

DeepWalk. DeepWalk [Perozzi *et al.*, 2014] is a typical NRL model that learns vertex representations based on network structures. We set parameters in DeepWalk as follows, window size $K = 5$, walks per vertex $\gamma = 80$ and representation dimension $k = 200$. For a vertex v, we take the representation $\mathbf{v}$ as network feature vector.

DeepWalk as Matrix Factorization. In previous part, we have introduced that DeepWalk can be trained in a matrix factorization form. Thus we factorize the matrix $M = (A + A^2)/2$ and take the factorized matrix X as representations of vertices.

2nd-LINE. LINE [Tang *et al.*, 2015] is another NRL model that learns network representations in large-scale networks. We employ the second-order proximity LINE (2nd-LINE) to learn representations for directed networks. Same as DeepWalk, we also set the representation length as 200.

5.3 5.3 Experimental Results and Analysis

Table 1, Table 2 and Table 3 show classification accuracies with different training ratios on different datasets. In these tables, we denote DeepWalk as DW , matrix factorization form of DeepWalk as MFDW and 2nd-LINE as LINE for short. We also show the performance of MMDW with η ranging from 10^{-4} to 10^{-2}. From these tables, we have the following observations:

(1) The proposed max-margin DeepWalk consistently and significantly outperforms all the baselines on all different datasets and different training ratios. Note that, MMDW achieves nearly 10% improvement on Citerseer and 5% improvement on Wiki when the training ratio is around 50%.

[2]https://people.cs.umass.edu/ mccallum/data.html

Table 1: Accuracy (%) of vertex classification on Cora.

%Labeled Nodes	10%	20%	30%	40%	50%	60%	70%	80%	90%
DW	68.51	73.73	76.87	78.64	81.35	82.47	84.31	85.58	85.61
MFDW	71.43	76.91	78.20	80.28	81.35	82.47	84.44	83.33	87.09
LINE	65.13	70.17	72.2	72.92	73.45	75.67	75.25	76.78	79.34
MMDW($\eta = 10^{-2}$)	**74.94**	**80.83**	**82.83**	**83.68**	**84.71**	**85.51**	**87.01**	**87.27**	**88.19**
MMDW($\eta = 10^{-3}$)	74.20	79.92	81.13	82.29	83.83	84.62	86.03	85.96	87.82
MMDW($\eta = 10^{-4}$)	73.66	79.15	80.12	81.31	82.52	83.90	85.54	85.95	87.82

Table 2: Accuracy (%) of vertex classification on Citeseer.

%Labeled Nodes	10%	20%	30%	40%	50%	60%	70%	80%	90%
DW	49.09	55.96	60.65	63.97	65.42	67.29	66.80	66.82	63.91
MFDW	50.54	54.47	57.02	57.19	58.60	59.18	59.17	59.03	55.35
LINE	39.82	46.83	49.02	50.65	53.77	54.2	53.87	54.67	53.82
MMDW($\eta = 10^{-2}$)	**55.60**	60.97	63.18	65.08	**66.93**	**69.52**	**70.47**	**70.87**	**70.95**
MMDW($\eta = 10^{-3}$)	55.56	**61.54**	**63.36**	**65.18**	66.45	69.37	68.84	70.25	69.73
MMDW($\eta = 10^{-4}$)	54.52	58.49	59.25	60.70	61.62	61.78	63.24	61.84	60.25

DeepWalk can not represent vertices in Citeseer and Wiki well, while MMDW is capable of managing such situation. These improvements demonstrate that MMDW is more robust and especially performs better when the quality of network representations is poor.

(2) What calls for special attention is that MMDW outperforms the original DeepWalk with half less training data on Citeseer and Wiki. It demonstrates that MMDW is effective when applied to predictive tasks.

(3) The proposed MMDW obtains considerable improvements than primal social representation learning methods. In contrast, the performance of original DeepWalk and DeepWalk as matrix factorization is unstable on various datasets. This indicates that the introduction of supervised information is important and MMDW is flexible to diversified networks.

Observations above demonstrate that MMDW is able to incorporate labeling information for generating high quality representations. MMDW is not application-specific. The learnt representations of vertices can be utilized in many tasks, including vertex similarity, link prediction, classification and so on. The idea of biased gradient can be easily extended to other matrix factorization methods.

5.4 5.4 Convergence and Parameter Sensitivity

MMDW optimizes the max-margin classifier and matrix factorization alternately. In the top part of Fig. 2, we show the convergence trend of accuracies when model is trained with different training ratios. We observe that our proposed MMDW always achieves the best performance after 2 or 3 iterations. The performance of MMDW rises rapidly at the beginning, then becomes stable. The fast convergence rate indicates the efficiency of training MMDW.

As the primal gradient and the bias are calculated through different ways, they are initially under different orders of magnitude. Thus, we introduce a hyper-parameter η to balance the biased gradient and original gradient. We fix training ratio to 50% and test the performance of MMDW with different η.

From the bottom part of Fig. 2, we observe that MMDW has a stable performance when η ranges from 10^{-5} to 10^{-2} and gains the best performance when $\eta = 10^{-2}$. A fixed parameter η is suitable for different datasets.

5.5 5.5 Visualization

In this paper, we propose max-margin DeepWalk to learn discriminative representations for social networks. To verify whether the learnt representations is discriminative, we show the 2D representations of vertices using t-SNE visualization tool in Fig. 3. In this figure, each dot represents a vertex and each color represents a category. We randomly select 4 categories to show the trend more clearly.

From Fig. 3, we observe that MMDW learns a better clustering and separation of the vertices. On the contrary, the representations learnt by DeepWalk tend to mix together. A well-separated representation is more discriminative and easier to categorize. These significant improvements prove the effectiveness of our discriminative learning model.

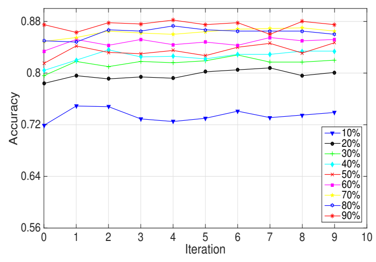
5.6 5.6 Case Study

To demonstrate the effectiveness of MMDW, we provide an instance in Cora dataset for example. The title of this instance is "Fast Online Q(λ)", and it belongs to the category of "Reinforcement Learning". As shown in Table 4, we list Top-5 nearest neighbors with the representations learnt by DeepWalk and MMDW. Here, we employ the cosine similarity to measure the distance between vertices.

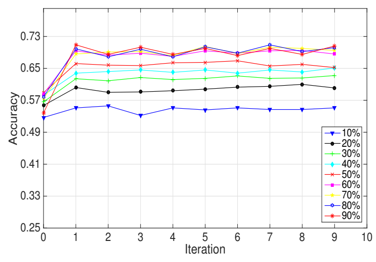
From Table 4, we observe that only 2 neighbors found by DeepWalk belong to the same category as the instance does, while MMDW finds 5. From the title of the instance, we can get that it is relevant to "online learning" and "Q problem". Most of the neighbors found by DeepWalk have no relatedness with these topics, while, most of the neighbors found by MMDW are closely related to them. It indicates that MMDW improves the quality of representations with the consideration of labeling information.

Table 3: Accuracy (%) of vertex classification on Wiki.

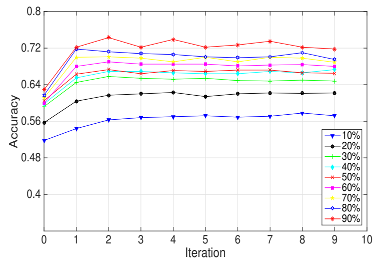
%Labeled Nodes	10%	20%	30%	40%	50%	60%	70%	80%	90%
DW	52.03	54.62	59.80	60.29	61.26	65.41	65.84	66.53	68.16
MFDW	56.40	60.28	61.90	63.39	62.59	62.87	64.45	62.71	61.63
LINE	52.17	53.62	57.81	57.26	58.94	62.46	62.24	66.74	67.35
MMDW($\eta = 10^{-2}$)	**57.76**	**62.34**	**65.76**	**67.31**	**67.33**	**68.97**	**70.12**	**72.82**	**74.29**
MMDW($\eta = 10^{-3}$)	54.31	58.69	61.24	62.63	63.18	63.58	65.28	64.83	64.08
MMDW($\eta = 10^{-4}$)	53.98	57.48	60.10	61.94	62.18	62.36	63.21	62.29	63.67



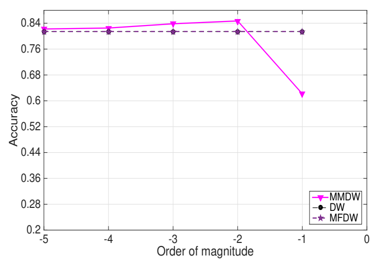
(a) Convergence on Cora



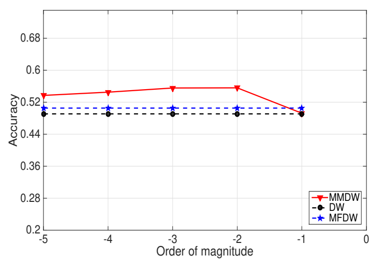
(b) Convergence on Citeseer



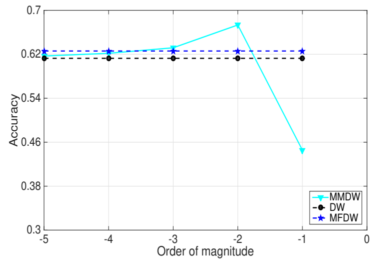
(c) Convergence on Wiki



(d) Parameter sensitivity on Cora



(e) Parameter sensitivity on Citeseer



(f) Parameter sensitivity on Wiki

Figure 2: Convergence and parameter sensitivity on different datasets

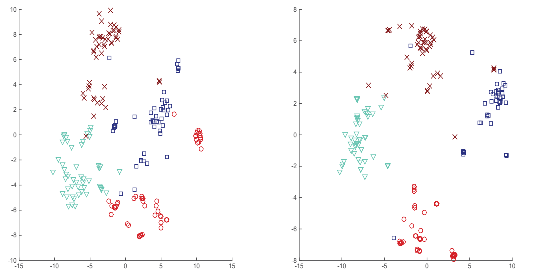


Figure 3: t-SNE 2D representations on Wiki (left: DeepWalk right: MMDW).

6 6 Conclusion and Future Work

In this paper, we propose max-margin DeepWalk (MMDW), a discriminative representation learning model for social networks. With the introduction of labeling information and max-margin principle, MMDW learns vertex representations which reflect both their network structure and labeling information. Experimental results on real-world datasets show that MMDW is effective for predictive tasks. Moreover, visualizations of the learnt representations confirm the discrimination of MMDW.

We will explore more in future work as follows:

- We have proved the effectiveness of max-margin DeepWalk. In the future, we aim to explore how to conduct max-margin methods on other social representation learning models, such as LINE.
- We learn a discriminative DeepWalk by transforming the orignal DeepWalk into matrix factorization form. In this way, it's easier to balance the bias vector and the gradient. Nevertheless, it's an offline method. In the future, we strive to explore the online discriminative learning methods.

7 Acknowledgements

This work is supported by the 973 Program (No. 2014CB340501), the Major Project of the National Social Science Foundation of China (13&ZD190) and the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

Table 4: Top-5 nearest neighbors by DeepWalk and MMDW

DeepWalk	
Title	Label
Truncating temporal differences On the efficient implementation of TD for reinforcement learning	Reinforcement Learning
Living in a partially structured environment How to bypass the limitation of classical reinforcement techniques	Neural Networks
Why experimentation can be better than perfect guidance	Theory
Averaged reward reinforcement learning applied to fuzzy rule tuning	Reinforcement Learning
A neural network pole balancer that learns and operates on a real robot in real time	Neural Networks

MMDW	
Title	Label
Applying online search to reinforcement learning	Reinforcement Learning
The efficient learning of multiple task sequences	Reinforcement Learning
A modular Q learning architecture for manipulator task decomposition	Reinforcement Learning
Truncating temporal differences On the efficient implementation of TD for reinforcement learning	Reinforcement Learning
Exploration and model building in mobile robot domains	Reinforcement Learning

References

[Blei *et al.*, 2003] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[Crammer and Singer, 2002] Koby Crammer and Yoram Singer. On the learnability and design of output codes for multiclass problems. *Machine learning*, 47(2-3):201–233, 2002.

[Hearst *et al.*, 1998] Marti A. Hearst, Susan T Dumais, Edgar Osman, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998.

[Keerthi *et al.*, 2008] S Sathiya Keerthi, Sellamanickam Sundararajan, Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. A sequential dual method for large scale multi-class linear svms. In *Proceedings of SIGKDD*, pages 408–416, 2008.

[McCallum *et al.*, 2000] Andrew McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval Journal*, 3:127–163, 2000.

[Mikolov *et al.*, 2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of ICIR*, 2013.

[Mikolov *et al.*, 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119, 2013.

[Pei *et al.*, 2014] Wenzhe Pei, Tao Ge, and Baobao Chang. Max-margin tensor neural network for chinese word segmentation. In *Proceedings of ACL*, pages 293–303, 2014.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of KDD*, pages 701–710, 2014.

[Roller, 2004] Ben Taskar Carlos Guestrin Daphne Roller. Max-margin markov networks. In *Proceedings of NIPS*, 2004.

[Sen *et al.*, 2008] Prithviraj Sen, Galileo Mark Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of WWW*, pages 1067–1077, 2015.

[Taskar *et al.*, 2004] Ben Taskar, Dan Klein, Michael Collins, Daphne Koller, and Christopher D Manning. Max-margin parsing. In *Proceedings of EMNLP*, volume 1, page 3, 2004.

[Yang *et al.*, 2015] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information. In *Proceedings of IJCAI*, 2015.

[Yu *et al.*, 2014] Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit Dhillon. Large-scale multi-label learning with missing labels. In *Proceedings of ICML*, pages 593–601, 2014.

[Zhu *et al.*, 2012] Jun Zhu, Amr Ahmed, and Eric P Xing. Medlda: maximum margin supervised topic models. *JMLR*, 13(1):2237–2278, 2012.