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 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

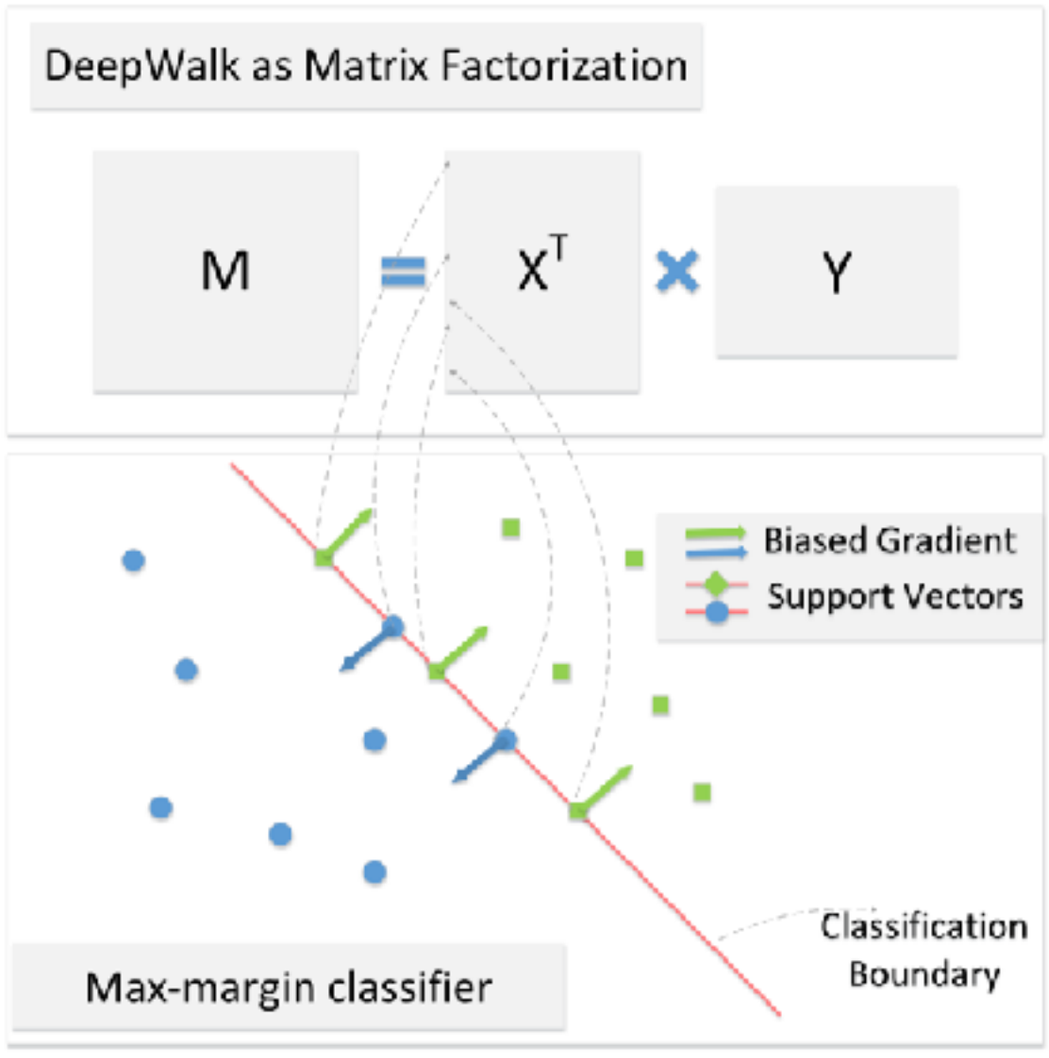
Introduction

- One-hot representation: due to its simplicity, such representation method has been widely adopted for network analysis. However, it usually suffers from the sparsity issue and does not fully consider relatedness between vertices.

- Distributed representation: network representation learning (NRL) is proposed to address these issues. 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.

Introduction



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.

Contributions

(1) We propose a discriminative NRL model, max-margin DeepWalk, to incorporate labeling information into vertex representations.

(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.

DeepWalk

Objective of DeepWalk:

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

The probability Pr(vj |vi) is computed using softmax function:

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

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

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

DeepWalk

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

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

where the factor λ controls the weight of regularization part.

Max-Margin DeepWalk

we take the learnt representations X as features and train an SVM for vertex classification. Suppose the training set is T ={(x1,l1),⋯,(xT,lT)}, the multi-class SVM aims to find optimal linear functions by solving the following constrained optimization problem:

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

where

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

$W = [w1,\cdots,wm]^T$ is the weight matrix of SVM, and $\xi = [\xi 1, \cdots, \xi T]$ is the slack variable that tolerates errors in the training set.

Max-Margin DeepWalk

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:

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

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:

$$\min_{\mathbf{Z}} \frac{1}{2}\|W\|_2^2 + \sum_{i=1}^{T}\sum_{j=1}^{m} e_i^j z_i^j \tag{8}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} z_i^j = 0, \quad \forall i$$

$$z_i^j \le C_{l_i}^j, \quad \forall i, j$$

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 $\mathbf{z}_i$.

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.

Optimization over X and Y

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

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

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.

Datasets

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 relation- ships 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.

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

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

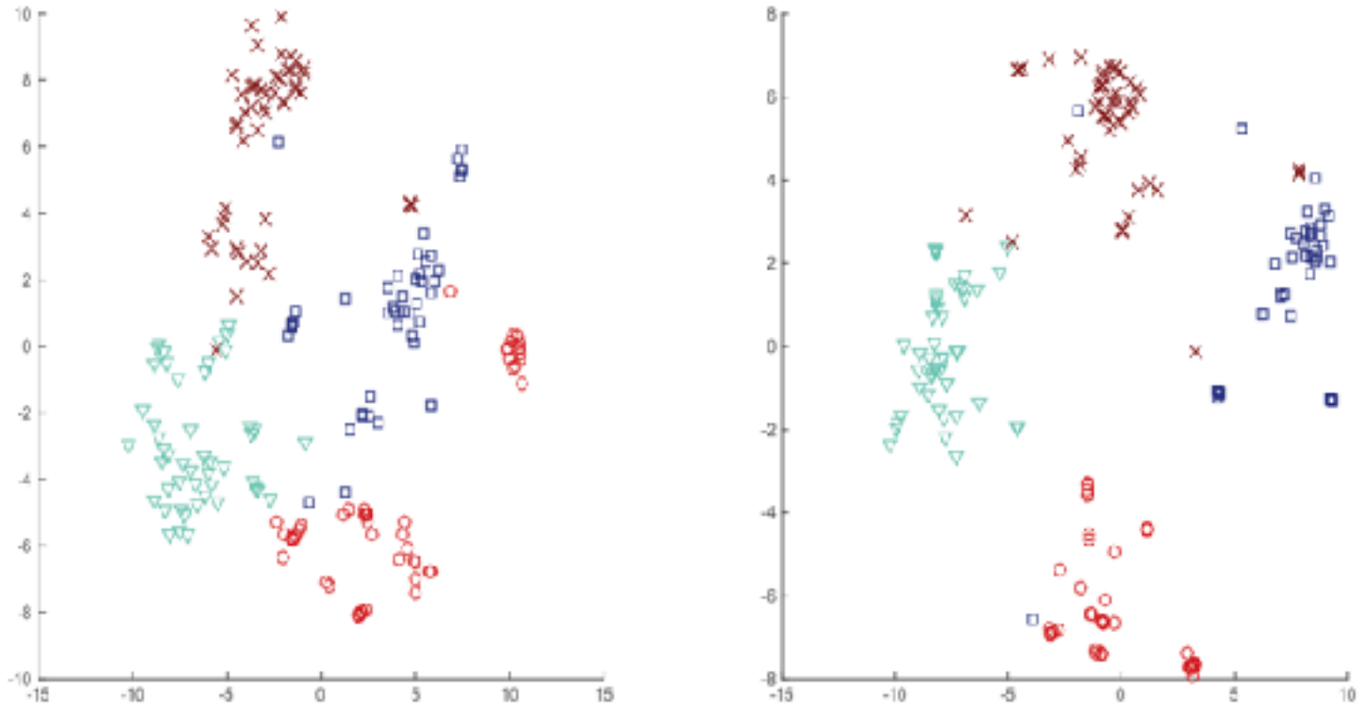
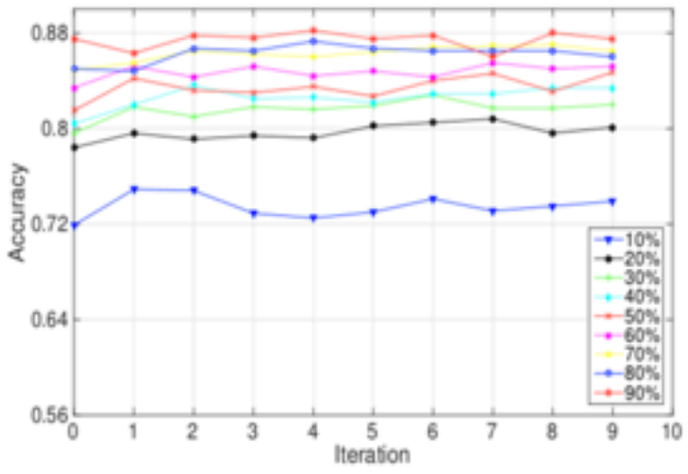
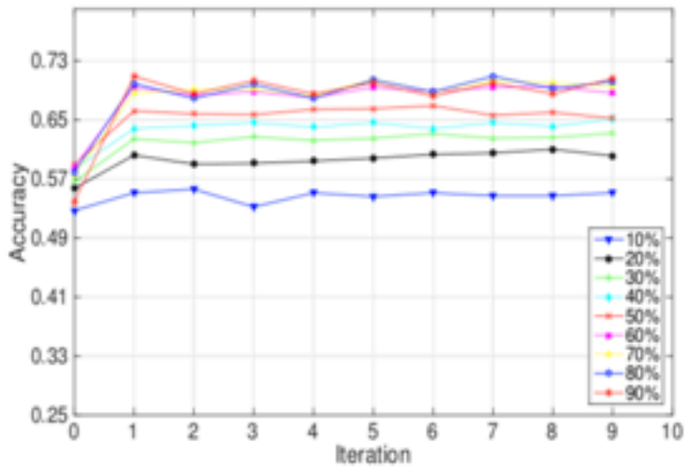


Figure 3: t-SNE 2D representations on Wiki (left: DeepWalk right: MMDW).

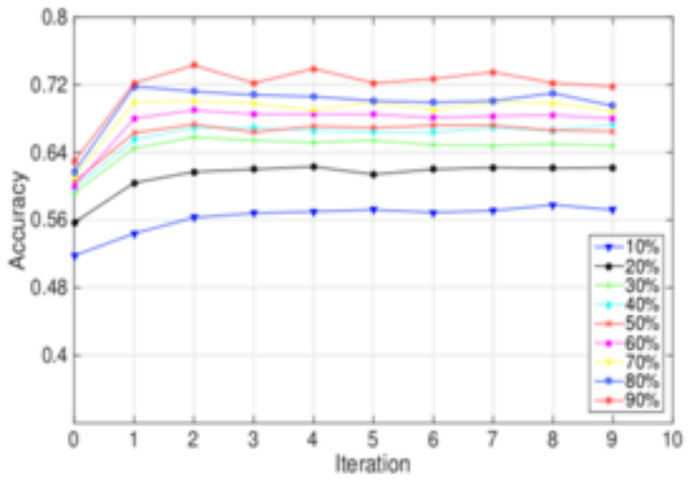
To verify whether the learnt representations is discriminative, we show the 2D representations of vertices using t-SNE visualization tool in Fig. 3. In this figure, each dot represents a vertex and each color represents a category



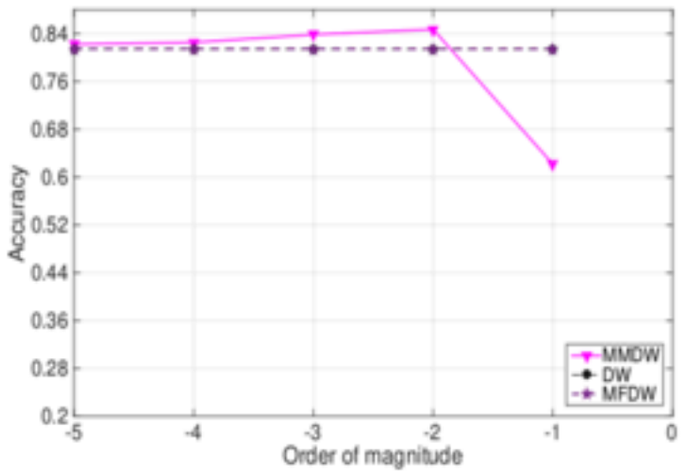
(a) Convergence on Cora



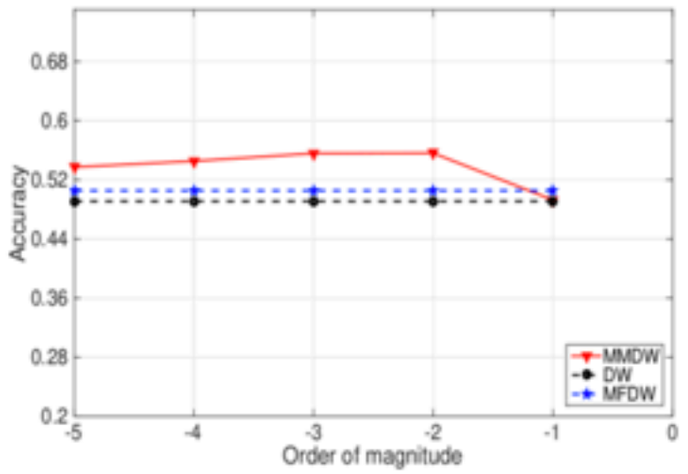
(b) Convergence on Citeseer



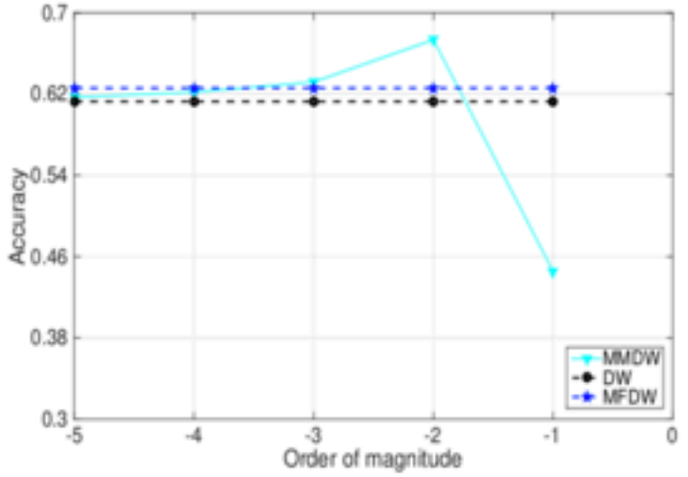
(c) Convergence on Wiki



(d) Parameter sensitivity on Cora



(e) Parameter sensitivity on Citeseer



(f) Parameter sensitivity on Wiki

Figure 2: Convergence and parameter sensitivity on different datasets

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