This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3039533, IEEE Transactions on Knowledge and Data Engineering

A Tensor-based Markov Chain Model for Heterogeneous Information Network Collective Classification

Chao Han, Jian Chen, Mingkui Tan, Michael K. Ng, Qingyao Wu*

Abstract—Heterogeneous Information Network (HIN) collecitve classification studies the problem of predicting labels for one type of nodes in a HIN which contains multiple types of nodes multiple types of links among them. Previous studies have revealed that exploiting relative importance of links is quite useful to improve node classification performance as connected nodes tend to have similar labels. Most existing approaches exploit the relative importance of links either by directly counting the number of connections among nodes or by learning the weight of each type of link from labeled data only. However, these approaches either neglect the importance of types of links to the class labels or may lead to overfitting problem. We propose a Tensor-based Markov chain (T-Mark) approach, which is able to automatically and simultaneously predict the labels for unlabeled nodes and give the relative importance of types of links that actually improve the classification accuracy. Specifically, we build two tensor equations by using the HIN and features of nodes from both labeled and unlabeled data. A Markov chain-based model is proposed and it is solved by an iterative process to obtain the stationary distributions. Theoretical analyses of the existence and uniqueness of such probability distributions are given. Extensive experimental results demonstrate that T-Mark is able to achieve superior performance in the comparison and obtain reasonable relative importance of links.

Index Terms—Heterogeneous information network, Node classification, relative importance of links, Tensor, Markov Chain, Iterative algorithm

1 Introduction

In recent years, Heterogeneous Information Network (HIN) [16] attracts a lot of attention. HIN is a complex network which contians multiple types of nodes and multiple types of links among these nodes. Collective classification [7] of nodes through the links among nodes, especially in a complex network, has attracted considerable attention in recent years and can be applied in many fields ranging from the Internet to social sciences, biology and many others [1], [2]. HIN is effective in representing structured data consisting of multiple links. For example, in research collaboration networks, research papers (as nodes) are published in the different areas and connected through citations (as links); in movie recommendation networks, movies (as nodes) with various genres are associated with each other in terms of directors or actors (as links); in social networks, users (as nodes) are interconnected by friendship networks with interesting topics (as links).

Node classification in HIN has been extensively studied [3], [4] in recent years, and has been successfully applied to many real-world applications, such as recommending specific items for individuals [1], discovering communities in social networks [2].

The most straightforward solution for this task is to decompose the HIN consisting of multiple types of links into a series of single-relational graphs [6]. Then, a set of classifiers are learned separately, and each for one single-relational graph. The decisions of the classifiers are combined to make a final prediction [7], [8]. However, such a solution neglects the relative importance of different types of links among nodes. Other approaches [9], [10] try to estimate the relative importance by tuning a weighting parameter for each types of links. Although such strategies can be effective in some cases, they may be prone to overfitting the training data, particularly when there are noise links and the training data is scant. Some other approaches [7], [11] try to exploit the relative importance of links by counting the number of connections among nodes through each type of links; such approaches are straightforward and easy to implement, but there are high risks that link counting may have only a weak discriminative effect for the classification purpose.

In this paper, we propose a new approach, called Tensor-based Markov chain (T-Mark), which is able to determine the class labels of nodes and obtain the ranking of types of links simultaneously. In T-Mark, we represent the HIN by a tensor which is a multi-dimensional array. To be more specific, we show an example of an authorship HIN over five authors in Fig. 1(a). One author links to the others with different types of links, such as "citation", "research area", "collaboration", and "same conference". The graph structure involves interconnected nodes with multiple types of links.

- C. Han is with the School of Electrical Engineering, Guangxi University, China. E-mail: hanchaoc@gmail.com.
- J. Chen, M. Tan and Q. Wu are with the School of Software Engineering, South China University of Technology. The Key Laboratory of Big Data and Intelligent Robot, Ministry of Education, China. E-mail: ellachen@scut.edu.cn, mingkuitan@scut.edu.cn, qyw@scut.edu.cn
- M. Ng is with the Department of Mathematics, The University of Hong Kong. E-mail: mng@maths.hku.hk.

Manuscript received November 4, 2020; revised June 28, 2020.

As shown in Fig. 1(b), a three-dimensional array is used to represent the relational dataset in Fig. 1(a). Each two-dimensional font slice represents an adjacency matrix for a single relation. The size of the tensor is $(5 \times 5 \times 4)$, and the (i, j, k) entry is nonzero if the ith node is connected to the jth node by using the kth relation.

The main contribution of this paper is to propose a method which can perform the relative importance of links and node classification simultaneously in HIN. To achieve this, we use a tensor to represent the HIN and construct transition probability graphs for the Markov model by using the tensor of relations and the description features of nodes. We also design an iterative algorithm to solve the tensor equations to obtain the stationary probability distributions w.r.t. nodes and relations for different classes. The basic idea is to imagine the nodes randomly walk in the transition probability graphs and transfer their label information to the connected neighbors. This leads to a new tensor-based Markov chain model calculating the stationary probability distributions of nodes and links as probability scores for node classification and relative importance of links .

Intuitively, for the node classification task, the top-ranked links tend to have a better discriminative effect for node classification. One hopes that relatively important type of links connecting the nodes with similar labels receives a high ranking value. In our approach, we predict the label of a node by 1) ordering a rank which respects the relative importance of links first and 2) performing label propagation with a random walk across the nodes based on their feature-based similarities and the relative importance of connecting links. To this end, we compute the rankings of types of links by 1) finding a confidence distribution for node classification first and 2) calculating the closeness among the large confidence nodes connected by this type of link.

The paper is based on our previous work by Han et al. [12]. Han et al. [12] proposed a Markov chain based model for node classification. This paper extends the previous work in the following directions:

- The tasks of node classification and link rankings in HIN are discussed jointly. An extensive discussion of the related work is also provided.
- We have extended the algorithm using an iterative mechanism to update the label matrix in each iteration to enhance the performance.
- The proofs of the existence and uniqueness of the stationary probability distributions of nodes and links for the proposed tensor based Markov chain model is presented.
- A computation complexity analysis of the proposed method is conducted.
- An example of synthetic HIN is used to show the computational procedure of the proposed T-Mark algorithm.
- The experimental evaluations are performed on a wider selection of datasets. A more detailed discussion of the effect of relative importance of links is provided. Parameters of the model are discussed and evaluated.

The rest of the paper is organized as follows. The related work is introduced in Section 2. The notations and tensor representation are described in Section 3. The proposed methodology is detailed in Section 4. Theoretical analysis is presented in Section 5. The experimental results are presented in Section 6. Conclusions are given in Section 7.

2 Related Work

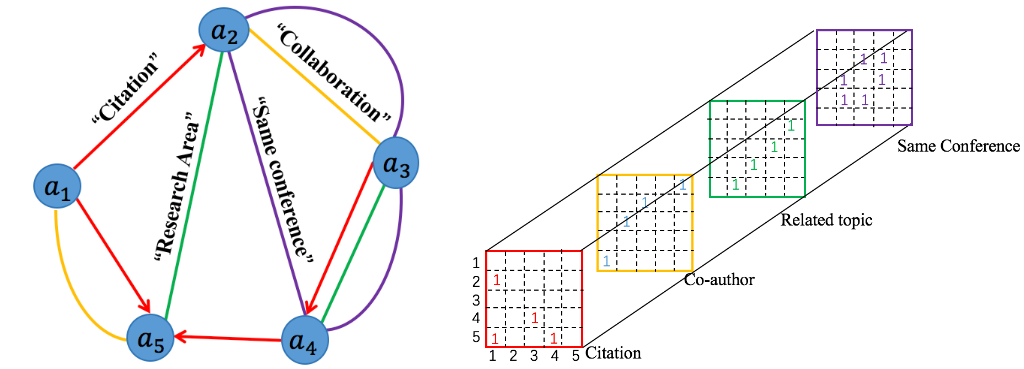
2.1 Collective Classification of HIN Nodes

Various approaches have been developed for classification in HIN data [7], [13]. Some methods adopt the straightforward strategy which decomposes the HIN classification problem into a number of independent classification tasks. Eldardiry and Neville [13] developed an iterative approach and incorporated the ensemble classifiers learned from heterogeneous networks into it to refine the prediction. Although these problem transformation approaches are simple and easy to implement because existing single-relational classification methods can be used directly as components. This method learns each type of link information sequentially and gradually reduce the prediction error. However, it is clear that this method is incapable of exploiting the relative importance of types of links.

Other methods considered a transformation from the relational information to the features. For example, Vijayan et al. [11] solved the multi-label collective classification in a multi-attribute HIN, where each relational view was transformed into a vector space by aggregating the label information. The counting number of connections among nodes through a relation is used as its weighting value. Obviously, the classification results are implicitly influenced by the weighting values of relations. Kong et al. [3] proposed an approach for solving this problem, which employs the meta-path method to transform a heterogeneous network to multiple relations. These relations are considered as a sequence of feature vectors by aggregating the label information of neighbors. This method is used as a baseline in our experiments. However, there is a high risk that the connection counting has only weak discrimination effect for the classification.

Another approaches were proposed to estimate the relation relevance by tuning the parameter values. For example, Jacob et al. [14] transformed the relations into a feature vector and then defined an edge function with multiple parameters for the relations. Shi et al. [9] developed an objective function for minimizing an empirical loss, which was solved by stochastic gradient boosting trees. In this approach, a set of weighting parameters was used for judiciously filtering out the data sources that were noisy. Satchidanand et al. [10] modeled multiple relations as a set of hypergraphs and performed parameter selection for these different hypergraphs. The deep learning methods also train many weights to model the neural networks. The compared Graph Inception [39] method in our experiments shows a decrease when the training data increases. It may be that too many weights will generate an overfitting problem.

Eswaran et al. [15] proposed ZooBP to perform on heterogeneous graphs with multiple types of nodes and links. Ji et al. [16] proposed an algorithm, RankClass, for both classification and ranking of the nodes in the heterogeneous network. This method assumed that the important node within each class played more important roles for classification. Our proposed T-Mark is different from these methods, where we aim to classify the nodes with feature descriptions and rank the connecting links simultaneously.



(a) (b)

Fig. 1. An example of authorship network. (a) Examples of multiple types of connections; (b) A tensor representation of the data.

2.2 Tensor-based Relational Learning

A tensor is a multi-dimensional array which is powerful and versatile to model HIN. Tensor-based relational learning techniques have been extensively used in many applications, such as community discovery [17], [18], link prediction [19], and node ranking [21].

A natural extension to learning multiple relations is to stack the matrices to be factorization and applying tensor factorization methods for analyzing multi-way interactions among nodes [19]. These approaches, which induce some inherent sharing of parameters between different nodes and links, have been applied successfully in many applications, such as community discovery [18], link prediction [20], and ranking [21].

Recently, Ng et al. [22] proposed a framework, MultiRank, to seek the stationary probability distributions of tensors for the ranking problem in HIN. Later on, this approach is employed for computing the hub, authority scores of an node, and the relevance scores of relations [23]. The approach is also used for discovering the community structure [24]. Subsequently, Li et al. [25] extend the method for image retrieval, where an image is represented by several visual concepts. Different from these approaches, we focus on the problem of node classification and relation ranking in which the node features and the labels information of each node are taken into account. A tensor-based transition probability graph with both labeled and unlabeled data is constructed. This leads to a semi-supervised learning approach for node classification and link ranking.

Various researchers have considered tensor factorization techniques to analyze multi-way interactions among nodes in HIN. For instance, Sun et al. [26] applied a 3-way Tucker decomposition on the user, query-term and web page data in order to personalize web search. Lin et al. [17] proposed a novel relational hypergraph representation for modeling multi-relational and multi-dimensional social data and studied an efficient factorization method for community extraction on a given meta-graph. Nickel et al. [19] proposed a three-way tensor factorization approach, RESCAL, which is a symmetric Tucker model, to exploit the collective learning effect. In these approaches, the number of decomposition in the tensor factorization is artificially selected, and the computation may not be unique.

3 Notations and Tensor Representation

We use a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to represent a HIN, where $\mathcal{V}$ denotes the multiple sets of nodes where each set includes one type of nodes, $\mathcal{E}$ denotes multiple sets of edges between any two nodes in $\mathcal{V}$, each corresponding to one type of links. Each edge connects two nodes with a specific type of link. Each node v_i is represented by a feature vector $\boldsymbol{f}_i \in \mathbb{R}^d$ and is associated with at least one labels. Suppose there are n nodes, m types of links, and q possible labels. The problem is to predict the class labels of unlabeled nodes in HIN with the relational information and attribute feature of nodes. Simultaneously, we obtain the rankings of links associated with each class label to analyze the relatively important links.

3.1 Tensor Representation

We use a three-way tensor to represent the multiple relations among nodes [22]. We call $\boldsymbol{\mathcal{A}}= (a_{i,j,k})$ a 3-dimensional tensor of size $(n \times n \times m)$, where $i, j = 1, 2, \ldots, n$, and $k = 1, 2, \ldots, m$. We refer (i, j) to be the indices for nodes and k to be the indices for relations. Specifically, if the ith node is linked to the jth node through the kth types of link, then $a_{i,j,k}$ is set to 1. $\boldsymbol{\mathcal{A}}$ is a nonnegative tensor for $a_{i,j,k} \geq 0, \forall i, j, k$.

In order to leverage the labeled and unlabeled nodes simultaneously for semi-supervised node classification and relation ranking, we construct a transition probability graph for a Markov chain of all the labeled and unlabeled nodes, and then make use of the idea in topic-sensitive PageRank [27] and random walk with restart [28] to propagate the label information from labeled data to unlabeled data.

In the graph, the Markov transition probabilities $\boldsymbol{\mathcal{O}}= (o_{i,j,k})$ and $\boldsymbol{\mathcal{R}}= (r_{i,j,k})$ w.r.t. nodes and links can be obtained by normalizing the entries of $\boldsymbol{\mathcal{A}}$ as follows:

$$o_{i,j,k} = \frac{a_{i,j,k}}{\sum\limits_{i=1}^{n} a_{i,j,k}}, \quad i = 1, \ldots, n, \tag{1}$$

$$r_{i,j,k} = \frac{a_{i,j,k}}{\sum\limits_{k=1}^{m} a_{i,j,k}}, \quad k = 1, \ldots, m. \tag{2}$$

Here, $o_{i,j,k}$ represents the probability of visiting the ith node by given that jth node is currently visited and the kth relation is used, and $r_{i,j,k}$ represents the probability of using the kth relation given that ith node is visited from the jth node.

Let $X_t = [X_t = 1, \dots, X_t = n]$ and $Z_t = [Z_t = 1, \dots, Z_t = m]$ be the random variables referring to visiting any particular node and using any particular type of links respectively at the time t. The transition probabilities can be written as follows:

$$o_{i,j,k} = P\left[X_t = i \,|X_{t-1} = j, Z_t = k\right],$$

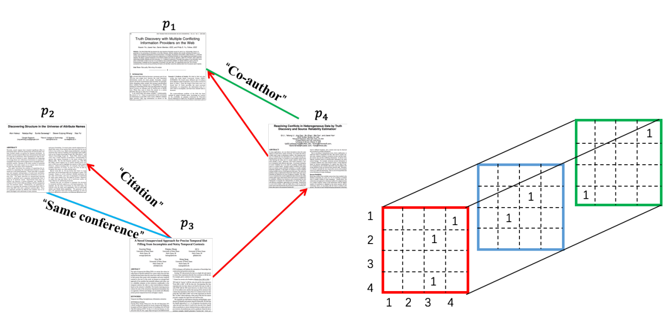
$$r_{i,j,k} = P\left[Z_t = k \,|X_t = i, X_{t-1} = j\right].$$

If there is a dangling node ($a_{i,j,k}$ is equal to 0 for all $1 \le i \le n$ [29]), the values of $o_{i,j,k}$ can be set to $1/n$ (an equal chance to visit any object). Similarly, $r_{i,j,k}$ can be set to $1/m$ (an equal chance to use any relation), if $a_{i,j,k}$ is equal to 0 for all $1 \le k \le m$. We call $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{R}}$ transition probability tensors which are the analog of transition probability matrices in Markov chains [30].

We assume that any two nodes in the HIN can be connected via some relations, so $\boldsymbol{\mathcal{A}}$ is irreducible. As we would like to determine the stable probability distributions of both nodes and links simultaneously, irreducibility is a reasonable assumption that we will use in the following analysis and discussion. It is clear that when $\boldsymbol{\mathcal{A}}$ is irreducible, the two corresponding tensors $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{R}}$ are also irreducible.

3.2 An Example

We select a subgraph from the DBLP bibliography network as an example (see Fig. 2(a)) to show the computational procedure of the proposed T-Mark algorithm. This HIN consists of 4 publications (as nodes), i.e., p_1, p_2, p_3 and p_4, and 3 types of relations, i.e. "co-author", "citation", and "same conference". For the "co-author" relation, the publications p_1 and p_2 have the same author "Jiawei Han". The publication p_3 cites the publications p_2 and p_4, and the publication p_4 cites the published p_1. The four publications are published at the TKDE 2008, WWW 2016, WWW 2019, and SIGMOD 2014. Thus the publications p_2 and p_3 are pulicated at the same conference WWW. We construct a tensor $\boldsymbol{\mathcal{A}}$ of size $(4 \times 4 \times 3)$ to represent this HIN (see Fig. 2(b)). Each front slice in the tensor corresponds to one relation in the HIN.



(a) (b)

Fig. 2. (a)An example of bibliography HIN with 3 types of relations; (b) A tensor representation for the HIN.

For the simplicity of calculating the tensors $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{R}}$, we give the 1-mode and 3-mode matricization of tensor $\boldsymbol{\mathcal{A}}$ which is denoted by $\boldsymbol{\mathcal{A}}_{(1)}$ and $\boldsymbol{\mathcal{A}}_{(3)}$. The size of matrix $\boldsymbol{\mathcal{A}}_{(1)}$ is 4×12, and the size of matrix $\boldsymbol{\mathcal{A}}_{(3)}$ is 3×16.

$$\boldsymbol{\mathcal{A}}_{(1)} = \begin{bmatrix} 0&0&0&1&0&0&0&0&0&0&0&1 \\ 0&0&1&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&1&0&0&0 \end{bmatrix},$$

$$\boldsymbol{\mathcal{A}}_{(3)} = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&1&0&1&1&\dots \\ 0&0&0&0&0&0&1&0&0&1&0&0&0&\dots \\ 0&0&0&1&0&0&0&0&0&0&0&0&1&\dots \end{bmatrix}.$$

We normalize the nodes of $\boldsymbol{\mathcal{A}}$ based on Eq. (1) and Eq. (2) to obtain $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{R}}$, respectively. Equivalently, we normalize each column of $\boldsymbol{\mathcal{A}}_{(1)}$ to obtain the tensor $\boldsymbol{\mathcal{O}}$ (see Fig. 3) and each column of $\boldsymbol{\mathcal{A}}_{(3)}$ to obtain the tensor $\boldsymbol{\mathcal{R}}$ (see Fig. 4).

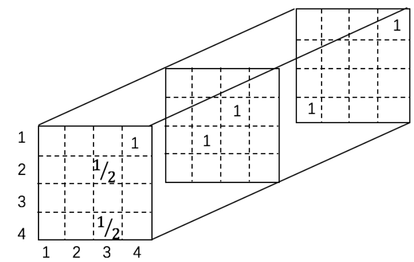


Fig. 3. An example of tensor $\boldsymbol{\mathcal{O}}$.

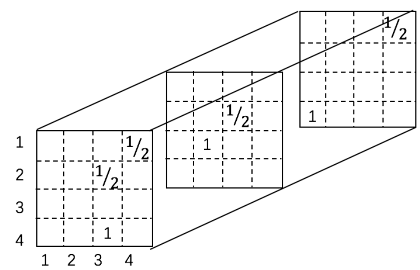


Fig. 4. An example of tensor $\boldsymbol{\mathcal{R}}$.

4 Methodology

In our approach, we consider a random walk on transition probability graphs. The graph is constructed by using the relation connections and feature-based similarities of both labeled and unlabeled data. It is a semi-supervised learning process for label propagation. In the HIN collective classification problem, the links among nodes can naturally be considered as the transition probability for label propagation. We proposed a tensor-based Markov chain model. An iterative algorithm is developed to solve a set of tensor equations of the proposed model to obtain the stationary distributions of nodes and types of links for classification and ranking, respectively.

4.1 Transition Probabilities from Relation Connections

Let $\boldsymbol{x}$ be a column vector of length n and $\boldsymbol{z}$ be a column vector of length m. Let $\boldsymbol{\mathcal{A}}\bar{\times}_1\boldsymbol{x}\bar{\times}_3\boldsymbol{z}$ be a vector in $\mathbb{R}^n$ such that

$$(\boldsymbol{\mathcal{A}}\bar{\times}_1\boldsymbol{x}\bar{\times}_3\boldsymbol{z})_i = \sum_{j=1}^{n}\sum_{k=1}^{m} a_{i,j,k}x_j z_k, \ i = 1, \dots, n.$$

Similarly, $\boldsymbol{\mathcal{A}}\bar{\times}_1\boldsymbol{x}\bar{\times}_2\boldsymbol{x}$ is a vector in $\mathbb{R}^m$ such that

$$(\boldsymbol{\mathcal{A}}\bar{\times}_1\boldsymbol{x}\bar{\times}_2\boldsymbol{x})_k = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{i,j,k}x_i x_j,\ k = 1, \ldots, m.$$

Given two transition probability tensors $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{R}}$, we study the following probabilities:

$$P\left[X_t = i\right] = \sum_{j=1}^{n}\sum_{k=1}^{m} o_{i,j,k} \times P\left[X_{t-1} = j, Z_t = k\right], \quad (3)$$

$$P\left[Z_t = k\right] = \sum_{i=1}^{n}\sum_{j=1}^{n} r_{i,j,k} \times P\left[X_t = i, X_{t-1} = j\right], \quad (4)$$

where $P\left[X_{t-1} = j, Z_t = k\right]$ is the joint probability distribution of X_{t-1} and Z_t, and $P\left[X_t = i, X_{t-1} = j\right]$ is the joint probability of X_t and X_{t-1}. Here we employ a product form of individual probability distributions for joint probability distributions. Under this assumption, (3) and (4) become

$$P\left[X_t = i\right] = \sum_{j=1}^{n}\sum_{k=1}^{m} o_{i,j,k} \times P\left[X_{t-1} = j\right] P\left[Z_t = k\right],$$

$$P\left[Z_t = k\right] = \sum_{i=1}^{n}\sum_{j=1}^{n} r_{i,j,k} \times P\left[X_t = i\right] P\left[X_{t-1} = j\right].$$

Here, we seek to achieve a stationary distribution of nodes, denoted by $\bar{\boldsymbol{x}} = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n]^T$ with $\sum_{i=1}^{n} \bar{x}_i = 1$ and a stationary probability distribution of relations, denoted by $\bar{\boldsymbol{z}} = [\bar{z}_1, \bar{z}_2, \ldots, \bar{z}_m]^T$ with $\sum_{k=1}^{m} \bar{z}_k = 1$, where

$$\bar{x}_i = \lim_{t\to\infty} P\left[X_t = i\right], \quad \text{and} \quad \bar{z}_k = \lim_{t\to\infty} P[Z_t = k]$$

for $1 \le i \le n$, and $1 \le k \le m$.

Using the above equations, we have

$$\bar{x}_i = \sum_{j=1}^{n}\sum_{k=1}^{m} o_{i,j,k}\bar{x}_j\bar{z}_k, i = 1, 2..., n, \quad (5)$$

$$\bar{z}_k = \sum_{i=1}^{n}\sum_{j=1}^{n} r_{i,j,k}\bar{x}_i\bar{x}_j, k = 1, 2, ..., m. \quad (6)$$

Formally, under the tensor operations for (5) and (6), we compute the stationary probabilities of the nodes and relations by solving the following two tensor equations:

$$\bar{\boldsymbol{x}} = \boldsymbol{\mathcal{O}}\bar{\times}_1\bar{\boldsymbol{x}}\bar{\times}_3\bar{\boldsymbol{z}}, \quad (7)$$

$$\bar{\boldsymbol{z}} = \boldsymbol{\mathcal{R}}\bar{\times}_1\bar{\boldsymbol{x}}\bar{\times}_2\bar{\boldsymbol{x}}. \quad (8)$$

4.2 Transition Probabilities from Node Features

We consider the case where the ith node is represented by a description feature of a d-dimensional vector $\boldsymbol{f}_i \in \mathbb{R}^d$. In our approach, we also construct a feature-based transition probability graph which employs the similarities among the nodes for the classification task. In order to compute the underlying node similarities, many distance metrics have been developed, such as Neighborhood Component Analysis (NCA) [31], Lare Marign Nearest Neighbor (LMNN) [32], Information-Theoretic Metric Learning (ITML) [33], cosine similarity, hamming distance.

Specifically, we compute the transition probabilities using cosine similarity which is given by

$$\cos\left(\boldsymbol{f}_i, \boldsymbol{f}_j\right) = \frac{\boldsymbol{f}_i \cdot \boldsymbol{f}_j}{\|\boldsymbol{f}_i\|\,\|\boldsymbol{f}_j\|},$$

where $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ are the feature vectors of the ith and jth nodes, respectively.

We construct an n-by-n matrix $\boldsymbol{C} = (c_{i,j})$ where $c_{i,j} = cos(\boldsymbol{f}_i, \boldsymbol{f}_j)$ indicates the cosine similarity between the ith and jth nodes. For each column, the sum of the transition probability matrix equals one. We obtain the transition probability matrix $\boldsymbol{W}$ by normalizing $\boldsymbol{C}$ w.r.t. each column, $\sum_{i=1}^{n} w_{ij} = 1, j = 1, 2, ..., n.$

We obtain the stable probability distributions of nodes on the feature similarities based transition probability graph by solving the following equation:

$$\bar{\boldsymbol{x}} = \boldsymbol{W}\bar{\boldsymbol{x}}, \quad (9)$$

with $\sum_{i=1}^{n} \bar{x}_i = 1$.

4.3 An Example

According to the synthetic example that is proposed in section 3.2, we obtain two transition tensors $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{R}}$. Suppose the cosine similarity matrix $\boldsymbol{C}$ for the nodes is given as

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix},$$

we can obtain the transition probability matrix $\boldsymbol{W}$ by normalizing each column of the matrix $\boldsymbol{C}$,

$$\boldsymbol{W} = \begin{bmatrix} 0.5 & 0 & 0 & 0.5 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0.5 \end{bmatrix}.$$

For this node classification task, there are two class labels, i.e., data mining (DM) and computer vision (CV). Assuming that p_1 and p_2 are labeled nodes assigned with DM and CV, respectively. The label assignment vector $\boldsymbol{l}$ for two class labels DM and CV $\boldsymbol{l} = [\boldsymbol{l}_{DM}, \boldsymbol{l}_{CV}]$ is given as

$$\boldsymbol{l} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

The task is to predict the class labels of unlabeled nodes p_3 and p_4. We initialize $\boldsymbol{x}$ and $\boldsymbol{z}$ as $\boldsymbol{x}_0^{DM} = [1, 0, 0, 0]^T$, $\boldsymbol{x}_0^{CV} = [0, 1, 0, 0]^T$, $\boldsymbol{z}_0 = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]^T$. By using Algorithm 1, we obtain the stationary probability distributions $\boldsymbol{x}^{DM}$, $\boldsymbol{x}^{CV}$, and $\boldsymbol{z}$ after finite iterations,

$$[\boldsymbol{x}^{DM}, \boldsymbol{x}^{CV}] = \begin{bmatrix} 0.90 & 0 \\ 0 & 0.90 \\ 0 & 0.10 \\ 0.10 & 0 \end{bmatrix},$$

$$[\boldsymbol{z}^{DM}, \boldsymbol{z}^{CV}] = \begin{bmatrix} 0.33 & 0.33 \\ 0.30 & 0.37 \\ 0.37 & 0.30 \end{bmatrix}.$$

Regarding $\boldsymbol{x}$, we can obtain the probabilities of an unlabeled node belonging to different classes. The ground-true classes for p_3 and p_4 are CV and DM. One can observe from $\boldsymbol{x}$ that the p_3 and p_4 has a larger probability to the class CV and DM, respectively. This observation demonstrated that our proposed method can achieve an accurate result for the node classification.

Moreover, we can obtain the relevance of a relation to different classes according to the stationary probability distribution of $\boldsymbol{z}$. One can see from Fig. 2(a) that the nodes p_1 and p_4 have one green undirected edge (i.e., the "co-author" relation) and one red directed edge (i.e., the "citation" relation). The probabilities of these two relations for DM class are 0.37 and 0.33, respectively, which are both higher than the blue edge (i.e., the "same conference" relation). This result shows that our method is able to provide a reasonable ranking for the relations.

4.4 The ICA-based T-Mark Algorithm

We propose a novel tensor-based Markove chain method, called T-Mark, to solve the node classification and relation ranking problem. The pseudo-code of the T-Mark algorithm is described in Algorithm 1. We start with a random walker on the labeled nodes. The walker iteratively visits the neighboring nodes with the transition probabilities $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{W}$ given in (7) and (9). At each step, it has probability $\alpha (0 < \alpha < 1)$ to return the label information of labeled nodes. We set a weighting parameter γ to scale the relation and feature information, which is denoted as $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{W}$, respectively. We use $\beta = \gamma \times (1-\alpha)$ to simplify the description of the equation. The walker with stationary probabilities will finally stay at different nodes. Formally, these stationary probabilities are computed using the following equation:

$$\bar{\boldsymbol{x}} = (1-\alpha-\beta)\boldsymbol{\mathcal{O}}\bar{\times}_1\bar{\boldsymbol{x}}\bar{\times}_3\bar{\boldsymbol{z}} + \beta\boldsymbol{W}\bar{\boldsymbol{x}} + \alpha\boldsymbol{l}, \tag{10}$$

where $\boldsymbol{l}$ is an assigned probability distribution vector of size n referring to the labeled nodes in the current label c. To construct $\boldsymbol{l}$, one simple way is to use a uniform distribution of the nodes with the class label $c (c = 1, 2, \ldots, q)$. More precisely,

$$[\boldsymbol{l}]_i = \begin{cases} 1/n_c, & \text{if } c \in Y_i; \\ 0 \quad , & \text{otherwise.} \end{cases} \tag{11}$$

where n_c is the number of nodes associated with the label c in the labeled dataset, Y_i is the label set of ith node.

ICA [7] is proposed for collective classification which is an iterative algorithm. The core idea of ICA is to update the feature representation by accepting the high confidence labels of unlabeled nodes in each iteration. We use this basic idea to update the start vector of labeled nodes l. We use the training data to set the vector l in the initialization. After each iteration of the Markov chain model, we update the vector l to accept some highly confident labels with high probability value in the prediction matrix. We set a relative threshold λ to control the acceptable label.

$$[\boldsymbol{l}]_i = \begin{cases} 1/n_l, & \text{if } c \in Y_i \text{ or } [\boldsymbol{x}]_i > \lambda; \\ 0 \quad , & \text{otherwise.} \end{cases} \tag{12}$$

where n_l is the number of nodes considering associated with the label c in the labeled dataset and with correponding probability value larger than the threshold λ.

In this paper, we present an iterative algorithm to solve the tensor Eq. (10) and (8) simultaneously. After finite iterations, we can obtain the stationary probability distributions of nodes and relations w.r.t. different class labels. We can obtain the top relevant relations which strongly related with a specific label based on the stationary probability values of relations w.r.t this class label. We can also use the stationary probability distributions of the nodes over different classes as the classification confident values to predict the labels for the unlabeled nodes.

Algorithm 1 The T-Mark Algorithm.

Input: $\boldsymbol{\mathcal{O}}$, $\boldsymbol{\mathcal{R}}$, $\boldsymbol{W}$, $\boldsymbol{l_0}$, $\boldsymbol{x}_0$, $\boldsymbol{z}_0$; Parameters: α, β, ε;
```
1: repeat
2:     set t = t + 1;
3:     if t > 2
4:     then Update l with equations(12);
5:     x_t = (1 − α − β) O ×̄_1 x_{t−1} ×̄_3 z_{t−1} + β W x_{t−1} + α l;
6:     z_t = R ×̄_1 x_t ×̄_2 x_t;
7: until ||x_t − x_{t−1}|| + ||z_t − z_{t−1}|| < ε;
```

4.5 Computational Complexity

The computational cost of T-Mark depends on the multiplication operation of tensor in the Algorithm 1. Since the tensor is the high-dimensional vector, we analyze the computational complexity of the tensor operation in step 2 and 3 (see Algorithm 1). The size of tensor $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{R}}$ both are $n \times n \times m$. The multiplication operation performs on each element of tensor. Thus the computational complexity of tensor operation is $O(n^2 m)$. However, the zero element of tensor does not perform the multiplication operation. Because the tensor is the representation of the relations among nodes, the tensor is sparse. We assume that there are D nonzero entries in $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{R}}$, the cost of the tensor calculations are of $O(D)$ arithmetic operations.

Let T be the iteration number of the proposed algorithm. The cost of obtaining the stationary distributions of x and z is $O(TD)$. For the classification, we need to get the stationary distributions of x and z for each label. Assume that we have q possible labels, the total cost of this algorithms is $O(qTD)$.

5 Theoretical Analysis

In this section, we show the existence and uniqueness of the stable probability distributions of $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{z}}$, and then conduct the convergence on the T-Mark algorithm.

Let $\Omega_n = \{\boldsymbol{x} = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n | x_i \geq 0, 1 \leq i \leq n, \sum_{i=1}^{n} x_i = 1\}$ and $\Omega_m = \{\boldsymbol{z} = (z_1, z_2, \cdots, z_m) \in \mathbb{R}^m | z_k \geq 0, 1 \leq k \leq m, \sum_{k=1}^{m} z_k = 1\}$. Also, let $\Omega = \{[\boldsymbol{x}, \boldsymbol{z}] \in \mathbb{R}^{n+m} | \boldsymbol{x} \in \Omega_n, \boldsymbol{z} \in \Omega_m\}$. Ω_n, Ω_m and Ω are closed convex sets.

Theorem 1. *Suppose $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{R}}$ are constructed in Section 4, $0 \leq \alpha, \beta < 1$, and $\boldsymbol{l} \in \Omega_n$ is given. For any $\boldsymbol{x} \in \Omega_n$ and $\boldsymbol{z} \in \Omega_m$, we have $(1-\alpha-\beta)\boldsymbol{\mathcal{O}}\bar{\times}_1\boldsymbol{x}\bar{\times}_3\boldsymbol{z} + \beta\boldsymbol{W}\boldsymbol{x} + \alpha\boldsymbol{l} \in \Omega_n$ and $\boldsymbol{\mathcal{R}}\bar{\times}_1\boldsymbol{x}\bar{\times}_2\boldsymbol{x} \in \Omega_m$.*

Proof.

$$(1-\alpha-\beta)\sum_{i=1}^{n}[\boldsymbol{\mathcal{O}}\bar{\times}_1\boldsymbol{x}\bar{\times}_3\boldsymbol{z}]_i + \beta\sum_{i=1}^{n}[\boldsymbol{W}\boldsymbol{x}]_i + \alpha\sum_{i=1}^{n}[\boldsymbol{l}]_i = 1,$$

$$\sum_{k=1}^{m}[\boldsymbol{\mathcal{R}}\bar{\times}_1\boldsymbol{x}\bar{\times}_2\boldsymbol{x}]_k = 1.$$

□

Using Theorem 1, we prove the existence of positive solutions to the tensor equations in (10) and (8) in Theorem 2.

Theorem 2. Suppose $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{R}}$ are constructed in Section 4. If $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{R}}$ are irreducible, then there exist $\bar{\boldsymbol{x}} \in \Omega_n$ and $\bar{\boldsymbol{z}} \in \Omega_m$ such that the following equalities hold:

$$(1-\alpha-\beta)\boldsymbol{\mathcal{O}}\bar{\times}_1\bar{\boldsymbol{x}}\bar{\times}_3\bar{\boldsymbol{z}}+\beta\boldsymbol{W}\bar{\boldsymbol{x}}+\alpha\boldsymbol{l}=\bar{\boldsymbol{x}}, \tag{13}$$

and

$$\boldsymbol{\mathcal{R}}\bar{\times}_1\bar{\boldsymbol{x}}\bar{\times}_2\bar{\boldsymbol{x}}=\bar{\boldsymbol{z}} \tag{14}$$

where $0<\alpha<1$, $0\le\beta<1$, and $\boldsymbol{l}\in\Omega_n$ is given. Moreover, both $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{z}}$ are positive.

Proof. We complete the proof by reducing the problem into a fixed point problem. We first define the following mapping $T:\Omega\to\Omega$

$$T([\boldsymbol{x},\boldsymbol{z}])=[(1-\alpha-\beta)\boldsymbol{\mathcal{O}}\bar{\times}_1\boldsymbol{x}\bar{\times}_3\boldsymbol{z}+\beta\boldsymbol{W}\boldsymbol{x}+\alpha\boldsymbol{l},\boldsymbol{\mathcal{R}}\boldsymbol{x}^2].$$

It is clear that $T([\boldsymbol{x},\boldsymbol{z}])\in\Omega)$ when $[\boldsymbol{x},\boldsymbol{z}]\in\Omega\ T$, and T is continuous. According to the Brouwer Fixed Point Theorem, there exists $[\bar{\boldsymbol{x}},\bar{\boldsymbol{z}}]\in\Omega$ such that $T([\bar{\boldsymbol{x}},\bar{\boldsymbol{z}}])=[\bar{\boldsymbol{x}},\bar{\boldsymbol{z}}]$, i.e., $(1-\alpha-\beta)\boldsymbol{\mathcal{O}}\bar{\times}_1\bar{\boldsymbol{x}}\bar{\times}_3\bar{\boldsymbol{z}}+\beta\boldsymbol{W}\bar{\boldsymbol{x}}+\alpha\boldsymbol{l}=\bar{\boldsymbol{x}}$ and $\boldsymbol{\mathcal{R}}\bar{\times}_1\bar{\boldsymbol{x}}\bar{\times}_2\bar{\boldsymbol{x}}=\bar{\boldsymbol{z}}$. Next we show that $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{z}}$ are positive. Suppose $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{z}}$ are not positive, i.e., there exist some entries of $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{z}}$ zero. Let $I=\{i|\bar{x}_i=0\}$, $J=\{j|\bar{x}_j=0\}$ and $K=\{k|\bar{z}_k=0\}$. Again I,J are proper subsets of $\{1,2,\cdots,n\}$ and K is a proper subset of $\{1,2,\cdots,m\}$. Let $\delta=\min\{\min\{\bar{x}_i|i\notin I\},\min\{\bar{z}_k|k\notin K\}\}$. We must have $\delta>0$. Since $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{z}}$ satisfy $\boldsymbol{\mathcal{O}}\bar{\times}_1\bar{\boldsymbol{x}}\bar{\times}_3\bar{\boldsymbol{z}}=\bar{\boldsymbol{x}}$, we have

$$(1-\alpha-\beta)\sum_{j=1}^{n}\sum_{k=1}^{m}o_{i,j,k}\bar{x}_j\bar{z}_k+\beta\sum_{j=1}^{n}w_{i,j}\bar{x}_j+\alpha l_i=\bar{x}_i=0, \quad \forall i\in I.$$

Let us consider the following inquantity:

$$\begin{aligned}&(1-\alpha-\beta)\delta^2\sum_{j\notin J}\sum_{k\notin K}o_{i,j,k}\\ \le&(1-\alpha-\beta)\sum_{j\notin J}\sum_{k\notin K}o_{i,j,k}\bar{x}_j\bar{z}_k\\ \le&(1-\alpha-\beta)\sum_{j=1}^{n}\sum_{k=1}^{m}o_{i,j,k}\bar{x}_j\bar{z}_k+\beta\sum_{j=1}^{n}w_{i,j}\bar{x}_j+\alpha l_i=0,\end{aligned}$$

for all $i\in I$. Hence we have $o_{i,j,k}=0$ for all $i\in I$ and for all $j\notin J$ for any fixed $k\notin K$. Thus the matrices $(o_{i,j,k})(k\notin K)$ are reducible. It implies that $\boldsymbol{\mathcal{O}}$ is reducible. By using the similar argument and considering the equation $\boldsymbol{\mathcal{R}}\bar{\times}_1\bar{\boldsymbol{x}}\bar{\times}_2\bar{\boldsymbol{x}}=\bar{\boldsymbol{z}}$, $\boldsymbol{\mathcal{R}}$ is also reducible. According to these results, we obtain a contradiction. Hence both $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{z}}$ must be positive. □

In [34], it has been given a general condition which guarantees the uniqueness of the fixed point in the Brouwer Fixed Point Theorem, namely, (i) 1 is not an eigenvalue of the Jacobian matrix of the mapping, and (ii) for each point in the boundary of the domain of the mapping, it is not a fixed point. In our case, we have shown in Theorem 2 that all the fixed points of T are positive when $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{R}}$ are irreducible, i.e., they do not lie on the boundary $\partial\Omega$ of Ω.

Theorem 3. Suppose $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{R}}$ are constructed in Section 4, and they are irreducible. If 1 is not the eigenvalue of $DT([\bar{\boldsymbol{x}},\bar{\boldsymbol{z}}])$ for all $[\bar{\boldsymbol{x}},\bar{\boldsymbol{z}}]\in\Omega/\partial\Omega$, then the solution vectors denoted by $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{z}}$, are unique.

According to Theorem 3, when $\boldsymbol{x}_t=\boldsymbol{x}_{t-1}$ and $\boldsymbol{z}_t=\boldsymbol{z}_{t-1}$ in T-Mark algorithm, then we obtain the unique solution vectors $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{z}}$ for $\boldsymbol{\mathcal{O}}\bar{\times}_1\boldsymbol{x}\bar{\times}_3\boldsymbol{z}=\boldsymbol{x}$ and $\boldsymbol{\mathcal{R}}\bar{\times}_1\boldsymbol{x}\bar{\times}_2\boldsymbol{x}=\boldsymbol{z}$. When $\boldsymbol{x}_t\neq\boldsymbol{x}_{t-1}$ and $\boldsymbol{z}_t\neq\boldsymbol{z}_{t-1}$, there exists a subsequence $[\boldsymbol{x}_s,\boldsymbol{z}_s]$ converging to $[\bar{\boldsymbol{x}},\bar{\boldsymbol{z}}]$ by using the fact that Ω is compact in $\mathbb{R}^{n+m}$. As we have shown that the solution vectors are unique, it implies that $[\boldsymbol{x}_t,\boldsymbol{z}_t]$ converges (up to a subsequence) to $[\bar{\boldsymbol{x}},\bar{\boldsymbol{z}}]$ which are the stationary probability vectors giving T-Mark values of nodes and types of links respectively for classification purpose effectively.

6 Experimental Results and Analysis

In this section, we conduct experiments to evaluate the performance of the proposed algorithm, we adopt several state-of-the-art methods as the baselines.

- ICA: It is commonly used as a comparison in collective classification [7]. For multiple types of links, we aggregate them all into one type of link for this algorithm.
- Hcc: Kong et al. [3] proposed this method for heterogeneous network classification, in which the meta path-based linkages among nodes can be viewed as multiple types of links.
- Hcc-ss: We employ a semi-supervised approach semiICA [8] to replace the base classifier ICA in Hcc.
- wvRN+RL [37]. It is a collective classification method which transfers content and structure information to the relationship among nodes, respectively. Hence, it can solve multi-relational classification problems.
- EMR: C. Preisach & L. Schmidi-Thieme [6] used an ensemble to combine multiple types of links while ignoring their differences. We train an ICA classifier for each type of link with SVM as the base classifier which votes for the final prediction.
- Highway Network (HN): Srivastava et al. [38] proposed a type of neural network layer that uses a gating mechanism to control the information flow through a layer. Stacking multiple highway layers allow for the training of deep networks.
- Graph Inception (GI): It designed a graph convolution-based model for learning the deep relational features in HINs and proposed the graph inception module to mix both complex and simple dependencies among the instances [39].

6.1 Performance Evaluation on DBLP Datasets

DBLP is reported by J. Ming et al. [35] which is extracted from DBLP[1] and contains publications from 20 computer science conferences on four research areas: database (DB), data mining (DM), artificial intelligence (AI), and information retrieval (IR). Each of them contains five conferences, see Table 1. Each publication of this dataset includes the authors, title, conference, and research area, besides the research area of all the authors and conferences. For this dataset, our task

1. http://dblp.uni-trier.de/

is to identify the interested area of each author based on their content and linkage information. For each author, a bag-of-words representation of all the publication titles published by the author is regarded as its content information. For connecness among authors, each conference is regarded as one type of links, and two authors have one type of link if they have published papers on the corresponding conference. Each author is assigned with a class label indicating his/her research area.

TABLE 1
The conferences of each research area in DBLP.

DB	DM	AI	IR
VLDB	KDD	IJCAI	SIGIR
SIGMOD	ICDM	AAAI	CIKM
ICDE	PAKDD	ICML	ECIR
EDBT	SDM	ECML	WWW
PODS	PKDD	CVPR	WSDM

In this experiment, we evaluate the performance of T-Mark algorithm on DBLP dataset and report the result in Table 3 We randomly pick up {10, 20, 30, 40, 50, 60, 70, 80, 90}% of the examples as the training data, and the remaining for testing. In order to randomize the experiments, for each given split, 10 test runs were conducted. The mean and standard deviation of the performance are reported in the table. The experimental results reveal that:

- Our algorithm always results in the best performance among all compared methods. Besides, the wvRN+RL method transforms the attribute feature to one type of link, and treat it equally with other linkage information. The results show that it is not competitive with our T-Mark method.
- Our T-Mark method outperforms the EMR and ICA approaches which both neglect the relative importance of links. This suggests the importance of the link relevance information in predicting the labels of nodes in the HIN.
- Both ICA and wvRN+RL methods without considering the semi-supervised learning mechanism suffer a performance degradation when there are less than 20% labeled data. Our method achieves significant improvement against these baselines in this case, which suggests the superiority of the idea of leveraging both labeled and unlabeled data in our algorithm.

Table 2 shows the top-5 conferences in each research area based on the relative importance of links results of our proposed method. The boldface ones indicate these conferences are also given in Table 1. Compared to Table 1, we can see that the top-4 conferences (oerstriking value) are included in the research areas based on the ranking result. This phenomenon is consistent with our expectation. For the other conferences that are not in top-5, PODS rank 6 in Database, PKDD ranks 6 in Data Mining, ECML ranks 6 and CVPR ranks 11 in Artificial Intelligence, WSDM ranks 19 in Information Retrieval. In general, the conference link ranking results have shown their topic preference of these four research areas. This observation reflects that the ranking results of our proposed method for the conferences of the DBLP dataset is reasonable.

TABLE 2
Top 5 conferences of each research area given by T-Mark.

DB	DM	AI	IR
VLDB	KDD	IJCAI	SIGIR
SIGMOD	ICDM	AAAI	CIKM
ICDE	PAKDD	ICML	ECIR
EDBT	SDM	ECML	WWW
CIKM	ICDE	SIGIR	IJCAI

6.2 Performance Evaluation on Movies

This Movie dataset is collected from IMDB [2] and Rotten Tomatoes [3], which is published by GroupLeans research group [4]. Each movie contains the tags, director, and genres. For this dataset, our task is to predict the genre for each movie based on their tags and director information. The tags are given by users, and we count the number of users give the same tag for one movie. A bag-of-word representation of these tags is considered as the content feature of movies. Each director is regarded as one type of links. If two movies are directed by the same director, they are related to each other. Each movie is assigned one of five genres, i.e., adventure, romance, thriller, war, documentary as its class label.

Table 4 shows the node classification accuracy of different methods on the Movies dataset. In the previous experiments, we have shown that the Hcc and Hcc-ss algorithms can get much better results than the other compared algorithms. In this experiment, we can see the EMR approach are the best. The reason is that each type of link is sparse. This dataset has a much larger scale and the connections among nodes are more sparse. For simplicity, we only select five classic types of movie genres to show the effect of the director links for predicting movie genre and ranking the directors in each genre.

We can see from Table 4 that the EMR algorithm gets the best performance. And the performance of our algorithm is better than the other algorithms. The reason is that the director links are too sparse. While the EMR algorithm aggregates all of the links, it leads to better performance than our proposed T-Mark algorithm. But the T-Mark algorithm also could get a better performance when even using only 20% training data compared with Hcc and Hcc-ss using 90% training data. Even though 90% of nodes are used as labeled data for training, the test accuracy is still undesirable, indicating that the director and the tag information from users are not sufficient for this task. Maybe more descriptions about the movie itself such as the story, video, music, costume, make-up, and, vision design, would increase the precision of the genre prediction.

Table 5 shows the top 10 directors of each movie genre. The ranking results is based on the director probability of each genre given by our proposed method. We can see that, for the 439 directors, they almost have different rankings in five genres, which infers that most directors prefer one specific type of movie. This result is consistent with our common sense that each director usually has one own style in productions. For example, the top 1 director of the documentary, Ivan Reitman, does not appear in other top 10. While some

2. http://www.imdb.com
3. https://www.rottentomatoes.com/
4. https://grouplens.org/datasets/hetrec-2011/

TABLE 3
The node classification accuracy on DBLP.

Percentage	T-Mark	TensorRrCc	GI	HN	Hcc	Hcc-ss	wvRN+RL	EMR	ICA
0.1	**0.928**	0.927	0.277	0.683	0.914	0.917	0.805	0.789	0.860
0.2	**0.933**	**0.933**	0.243	0.725	0.924	0.927	0.876	0.818	0.919
0.3	**0.935**	**0.935**	0.267	0.753	0.929	0.929	0.880	0.835	0.922
0.4	**0.935**	**0.935**	0.304	0.770	0.930	0.929	0.888	0.847	0.927
0.5	**0.939**	0.938	0.436	0.787	0.932	0.932	0.898	0.855	0.928
0.6	**0.939**	0.938	0.410	0.790	0.934	0.933	0.901	0.858	0.928
0.7	**0.940**	0.939	0.464	0.793	0.935	0.934	0.904	0.863	0.929
0.8	**0.940**	**0.940**	0.489	0.806	0.935	0.935	0.904	0.865	0.933
0.9	**0.940**	**0.940**	0.575	0.803	0.937	0.938	0.908	0.860	0.933

TABLE 4
The node classification accuracy on Movies.

Percentage	T-Mark	TensorRrCc	GI	HN	Hcc	Hcc-ss	wvRN+RL	EMR	ICA
0.1	0.441	0.441	0.309	0.453	0.435	0.426	0.318	**0.486**	0.203
0.2	0.483	0.483	0.297	0.483	0.456	0.453	0.318	**0.537**	0.219
0.3	0.511	0.511	0.292	0.506	0.460	0.458	0.309	**0.569**	0.239
0.4	0.518	0.518	0.302	0.531	0.461	0.460	0.308	**0.582**	0.238
0.5	0.529	0.529	0.348	0.543	0.467	0.468	0.309	**0.600**	0.254
0.6	0.546	0.546	0.299	0.563	0.473	0.471	0.306	**0.613**	0.258
0.7	0.549	0.549	0.391	0.572	0.478	0.476	0.314	**0.612**	0.257
0.8	0.553	0.553	0.376	0.579	0.474	0.473	0.300	**0.613**	0.258
0.9	0.560	0.560	0.339	0.594	0.491	0.486	0.303	**0.629**	0.268

famous directors such as Alfred Hithchcok, Joel Schumacher, Akira kurosawa, and Steven Spielberg, have many works with different styles.

6.3 Link Selection Analysis

HIN is a complex network which contains many useless links. In this section, we will conduct the experiment on the NUS dataset to demonstrate the importance of link selection.

NUS is a real-world image dataset collected by Lab for Media Search in National University of Singapore (NUS) [5]. More details of this dataset can be found in [36]. For this paper, we choose two high-level concepts, i.e., scene and object, as the label of 5780 images. A bag-of-words of SIFT description of length 500 is regarded as the feature of each image. These images are correlated with each other through 41 tags given by users. Our task is to identify these images are pure scenes or include one noticeable object.

In this experiment, we evaluate the effect of relative importance of links to nodes classification. If a link has a large probability of connecting the nodes belonging to the same class label, this link is considered as a relevant link. Otherwise, it is an irrelevant link. The relevant links are able to contribute more than irrelevant links in the process of label propagation for accurate and efficient nodes classification. For the NUS dataset, most of the images (as nodes) are linked by multiple tags (as links). If most of the images linked by a specific tag have the same class, this tag is considered as a relevant tag to this class label, verse visa.

In this experiment, we select two sets of tags (as link sets) from 1,000 tags. For the first link set, namely Tagset1, we select the tags which have at least 6% images are linked by this tag, and rank these tags based on their probabilities of connecting nodes belonging to the same class. The top 41 links with the largest probability are selected and listed in Table 6.

For the second link set, namely Tagset2, we rank 1,000 tags based on their frequency of appearance in the HIN. the top 41 tags with the largest frequency are selected and listed in Table 7.

We construct two HIN using the links in Tagset1 and Tagset2, and compared the performances of T-Mark on these two HIN. For each HIN, we randomly pick up {10, 30, 50, 70, 90}% of the nodes as labeled data, and use the remains as unlabeled data. Table 8 shows the mean of the accuracy of T-Mark over 10 trials on these two HIN using Tagset1 and Tagset2. We can see that T-Mark achieves a high accuracy on the HIN with Tagset1. The accuracy is 0.955 even with only 10% labeled data. Whilst the accuracy of T-Mark is only 0.692 even though 90% images are used as labeled data in the HIN with Tagset2.

We randomly select two classes ("Scene" and "Object") in the NUS dataset, and compare the top-ranked links in Tagset1 and Tagset2 w.r.t. these two classes. Tables 9 and 10 show the top 12 links in the Tagset1 and Tagset2 link set, respectively. For the tags in Tagset1 (see Table 9), one can see that the tags top-ranked w.r.t. "Scene" and "Object" are quite different. Meanwhile, these tags are relevant to the semantics of these two classes.

For the tags in Tagset2 (see Table 10), the top-6 tags w.r.t. the "Scene" and "Object" classes are similar, only have a small difference in orders. These tags may have weak effects on discriminating the "Scene" class and the "Object" class.

6.4 Multi-label Collective Classification of HIN on ACM

In this section, we conduct an experiment on a multi-label dataset ACM to evaluate the performance of T-Mark

5. http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm

TABLE 5
Top 10 directors of each movie genre.

Ranking	Adventure	Documentary	Romance	Thriller	War
1	Akira Kurosawa	Ivan Reitman	Alfred Hitchcock	Alfred Hitchcock	Alfred Hitchcock
2	Joel Schumacher	Akira Kurosawa	Clint Eastwood	Steven Spielberg	Howard Hawks
3	William Wyler	Woody Allen	Steven Spielberg	Clint Eastwood	John Badham
4	Renny Harlin	Martin Scorsese	Joel Schumacher	Roger Donaldson	Wes Craven
5	George Miller	Sydney Pollack	Werner Herzog	Woody Allen	Peter Howitt
6	Oliver Stone	Stephen Hopkins	Akira Kurosawa	Brian De Palma	Michael Mann
7	John Huston	John Woo	Ron Howard	Richard Fleischer	Oliver Hirschbiegel
8	Phillip Noyce	Ethan Coen	Don Siegel	Michael Apted	Jim Gillespie
9	Billy Wilder	Sidney Lumet	Terry Gilliam	William Wyler	Christian Duguary
10	Peter Jackson	John Sturges	Kenneth Branagh	Renny Harlin	Steven Spielberg

TABLE 6
The tags in Tagset1 (each tag corresponding to one types of links).

No.	Tag			
1 − 4	sky	water	clouds	landscape
5 − 8	sunset	architecture	portrait	reflection
9 − 12	animal	building	animals	lake
13 − 16	mountains	cute	abandoned	grass
17 − 20	mountain	window	cat	sunrise
21 − 24	zoo	bridge	cloud	dog
25 − 28	fall	face	square	rain
29 − 32	airplane	eyes	home	cold
33 − 36	windows	sign	flying	plane
37 − 40	arizona	manhattan	peace	rural
41	sports			

TABLE 7
The tags in Tagset2 (each tag corresponding to one types of link).

No.	Tag			
1 − 4	nature	sky	blue	water
5 − 8	clouds	red	green	bravo
9 − 12	landscape	explore	sunset	white
13 − 16	night	architecture	portrait	city
17 − 20	travel	trees	california	reflection
21 − 24	animal	girl	interestingness	building
25 − 28	river	animals	lake	abandoned
29 − 32	window	cat	sunrise	zoo
33 − 36	bridge	dog	baby	buildings
37 − 40	food	storm	moon	skyline
41	cats			

TABLE 8
The node classification accuracy on Movies.

Percentage	Tagset1	Tagset2
0.1	0.955	0.664
0.2	0.954	0.672
0.3	0.958	0.683
0.4	0.956	0.684
0.5	0.959	0.682
0.6	0.959	0.692
0.7	0.960	0.688
0.8	0.959	0.686
0.9	0.961	0.692

TABLE 9
Top-12 tags in Tagset1 given by T-Mark.

Scene	sky	clouds	sunset	water
	landscape	architecture	reflection	abandoned
	bridge	lake	building	window
Object	portrait	cat	animals	face
	animal	zoo	rain	sports
	dog	cute	airplane	fall

TABLE 10
Top-12 tags in Tagset2 given by T-Mark.

Scene	sky	clouds	sunset	water
	landscape	bravo	portrait	architecture
	blue	nature	reflection	abandoned
Object	sky	clouds	landscape	bravo
	water	sunset	nature	cat
	portrait	animals	animal	baby

algorithm on multi-label classification.

ACM dataset is extracted from the ACM digital library [6] with KDD conferences from 1999 to 2010 and SIGIR conferences from 2000 to 2010, which has been published by [22]. Each publication contains the title, keywords, authors, concepts, conference, citations, published year, and index terms. The index terms of papers are given by ACM based on the ACM Computing Classification Systems [7]. The task of this experiment is to predict the index term for each publication based on their representation and links. Each publication is represented by a bag-of-words vector from title terms. The other information implicates the correlation among publications, such that they can be organized into six types of links, i.e., authors, concepts, conferences, keywords, published year and citations. Note that the authors, concepts, conferences, keywords, and published year are non-directed. For example, if two publications have one same keyword, we consider there are two converse links with different start-end nodes. While the citation link is directed, we just confirm that the publication is related to the cited paper.

Fig. 5 shows the relative importance of links w.r.t. each class labels based on the ranking of types of links. The larger the probability value of the type of link on one class, this link is more important w.r.t. this class. We can see the probability distributions of link types over different classes are similar.

6. http://dl.acm.org/
7. https://dl.acm.org/ccs/ccs.cfm

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3039533, IEEE Transactions on Knowledge and Data Engineering

Two types of links (”concept” and ”conference”) are more important than others, which means that many nodes that are interconnected through these two types of links have similar class labels.

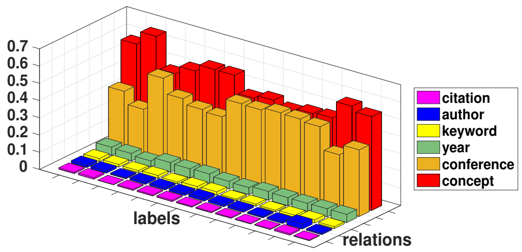


Fig. 5. The relative importance of link types on ACM given by T-Mark.

In this experiment, we predict the index terms for the publications on the ACM dataset. One publication is related to different information, such as authors, conference, keywords, published year, citations, and concepts. The classification result on ACM is shown in Table 11. From the table, we can see our algorithm outperforms the compared methods with different percentages of labeled data. Specifically, when the percentage of labeled data is less than 30%, our algorithm has a great superiority. The EMR and wvRN+RL algorithms have poor performances, and similar observation can also be found in the experiments on DBLP. The reason is that these algorithms treat all types of links equally.

6.5 Parameters Selection

In this part, we discuss the parameters and the convergence of the T-Mark algorithm. The proposed algorithm has two essential parameters, i.e., the restart parameter α and the scale parameter γ.

The parameter α indicates the importance of supervision information at each iteration of our semi-supervised learning model. In general, when α has a larger value, the algorithm performs better. We test the performance of the algorithm on two datasets when α varies from 0.1 to 0.99. From Fig. 6, we can see the accuracy firstly increases and then goes down varying with α increasing. It gets best when $\alpha = 0.8$ on DBLP and we set $\alpha = 0.8$ as the default value in all experiments. The value of α controls how much information we use from the labeled data. When α is larger than 0.8, the information from labeled data can not increase the accuracy. It is necessary learning from the content feature and relational information of nodes themselves.

Some differences appear in NUS dataset. The accuracy keeps going up with α varying from 0.1 to 0.99. And when α is larger than 0.6, the increment has a little drop. In the experiment, we set $\alpha = 0.9$ as the default value.

For the other two datasets, ACM and Movies, they show the same trend on the NUS dataset, and we set $\alpha = 0.9$ as the default value for them in the experiments.

Parameter γ mostly depends on the effect of the description features and relational information for the classification accuracy. When $\gamma = 0$, it means only use the relational information. When $\gamma = 1$, it means only use the feature

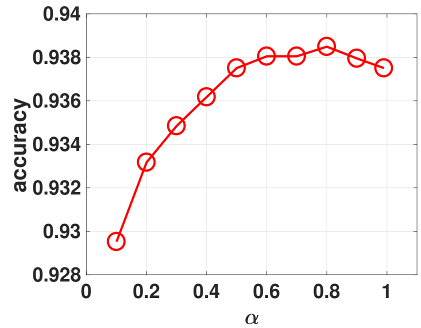


Fig. 6. The accuracy of T-Mark vs. parameter α on DBLP.

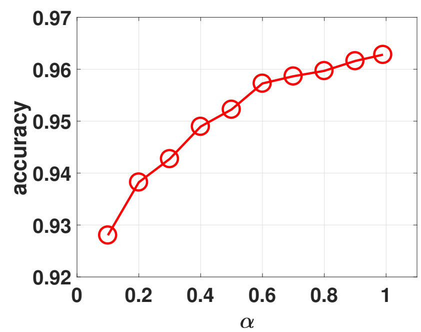


Fig. 7. The accuracy of T-Mark vs. parameter α on NUS.

information. When γ is larger than zero, it means the relational information has a positive effect on the classification result. And when γ is larger, the feature takes more roles in the classification process. For different datasets, they have a different tendency which we will discuss more in the following experiments.

We first test the performance of the algorithm on DBLP when γ varies from 0 to 1. From Fig. 8, we can see the variance of accuracy when γ increases. We get the worst result when only use the feature information, which is lower than 0.8. Though the performance achieves over than 0.9 when only using the relational information, the result is better when using both relational and feature information. The figure shows that our algorithm performs best when $\gamma = 0.6$ on DBLP, and we set it as the default value.

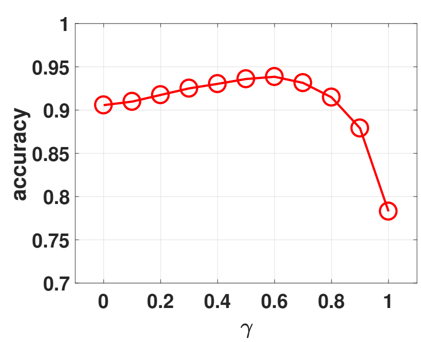


Fig. 8. The accuracy of T-Mark vs. parameter γ on DBLP.

We conduct the same experiment on NUS dataset. we can see from Fig. 9 that it shows the similar tendency with the

TABLE 11
The node classification performance under Macro F1 on ACM.

Percentage	T-Mark	TensorRrCc	GI	HN	Hcc	Hcc-ss	wvRN+RL	EMR	ICA
0.1	**0.940**	**0.940**	0.220	0.618	0.430	0.569	0.105	0.265	0.049
0.2	0.966	**0.968**	0.528	0.729	0.478	0.912	0.115	0.340	0.048
0.3	0.978	**0.988**	0.655	0.722	0.559	0.953	0.157	0.377	0.105
0.4	0.989	**0.993**	0.725	0.739	0.855	0.988	0.173	0.408	0.194
0.5	0.992	**0.997**	0.734	0.756	0.972	0.995	0.180	0.433	0.570
0.6	0.995	**0.997**	0.816	0.756	0.991	0.995	0.180	0.434	0.860
0.7	0.995	**0.997**	0.821	0.758	0.995	0.996	0.180	0.469	0.947
0.8	0.995	**0.997**	0.659	0.773	0.995	0.995	0.180	0.460	0.989
0.9	0.995	**0.997**	0.658	0.785	0.996	**0.998**	0.179	0.451	0.987

result on DBLP dataset. However, when γ varies from 0 to 0.4, the line keeps stable. Then it goes down when γ increases. It shows that it's enough to get the best result when only using the tag information in NUS dataset. And some little representation information cannot affect the classification result. When the representation information takes much more role in the process, the accuracy will drop. We set $\gamma = 0.4$ as the default value in all the experiments.

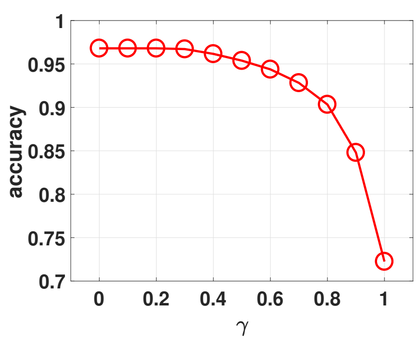


Fig. 9. The accuracy of T-Mark vs. parameter γ on NUS.

6.6 Convergence Study

In this section, we discuss the convergence of the T-Mark algorithm. We show the difference of neighboring iteration ($\rho = \|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\| + \|\boldsymbol{z}_t - \boldsymbol{z}_{t-1}\|$) w.r.t. the iteration number on four datasets. The result is shown in Fig. 10. We can see from this figure that the difference drops to zero or keeps stable when the iteration number is larger than 10, which means the algorithm has good convergence.

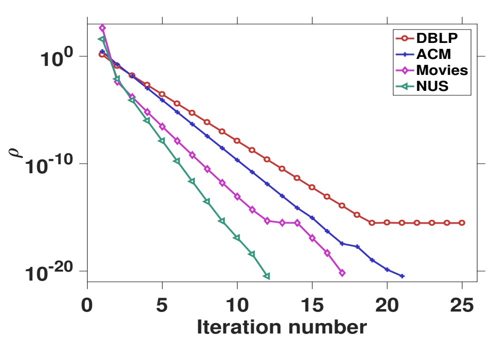


Fig. 10. The convergence curve of T-Mark on four datasets.

7 Conclusion

In this paper, we have proposed a novel algorithm, called T-Mark, for node classification in the Heterogeneous Information Network (HIN). The relative importance of the link types referring to class labels can improve classification performance. We used a three-dimensional tensor to represented the HIN and introduced a scheme to determine the rankings of link types and the class labels of nodes simultaneously based on the probability distributions. To solve the tensor equations, we develop an iterative algorithm based on the Markov chain to obtain the stationary probability distributions of nodes and types of links.Moreover, theoretical analyses were provided to declare the existence and the uniqueness of the stationary probability distributions. Extensive experiments on four real-world datasets demonstrate the superiority of the proposed method in node classification and its effectiveness in link rankings.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China Projects No. 62072186, No.61876208, the Hong Kong Research Grant Council General Research Fund No.12200317, No. 12300218, No. 12300519, No. 17201020, the Key-Area Research and Development Program of Guangdong Province No. 2018B010108002, the Guangdong Basic and Applied Basic Research Foundation No. 2019B1515130001, the Guangdong Special Branch Plans Young Talent with Scientific and Technological Innovation No. 2016TQ03X445, the Pearl River ST Nova Program of Guangzhou No. 201806010081, the Guangzhou Science and Technology Planning Project No. 201904010197, the Opening Project of Guangdong Key Laboratory of Big Data Analysis and Processing, the Support Program for Postdoctoral Innovative Talents of Guangxi.

References

[1] D. Le-Phuoc, H. N. M. Quoc, H. N. Quoc, T. T. Nhat, and M. Hauswirth, "The graph of things: A step towards the live knowledge graph of connected things," Web Semantics: Science, Services and Agents on the World Wide Web, vol. 37, pp. 25–35, 2016.

[2] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," Semantic web, vol. 8, no. 3, pp. 489–508, 2017.

[3] X. Kong, P. S. Yu, Y. Ding, and D. J. Wild, "Meta path-based collective classification in heterogeneous information networks," in Proc. of the 21st ACM Int. Conf. on Information and Dnowledge Management, 2012, pp. 1567–1571.

[4] Y. Zhang, Y. Xiong, X. Kong, and Y. Zhu, "Netcycle: Collective evolution inference in heterogeneous information networks."

[5] S. A. Kale and S. Chaurasia, "Different approaches for ranking fraud detection for mobile apps application: An analysis of comparative study on different algorithms," 2017.

[6] C. Preisach and L. Schmidt-Thieme, "Ensembles of relational classifiers," Knowledge and Information Systems, vol. 14, no. 3, pp. 249–272, 2008.

[7] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," AI magazine, vol. 29, no. 3, p. 93, 2008.

[8] L. McDowell and D. Aha, "Semi-supervised collective classification via hybrid label regularization," arXiv preprint arXiv:1206.6467, 2012.

[9] X. Shi, J.-F. Paiement, D. Grangier, and S. Y. Philip, "Learning from heterogeneous sources via gradient boosting consensus." in SDM, 2012, pp. 224–235.

[10] S. N. Satchidanand, H. Ananthapadmanaban, and B. Ravindran, "Extended discriminative random walk: a hypergraph approach to multi-view multi-relational transductive learning," in Proc. of the 24th Int. Conf. on Artificial Intelligence, 2015, pp. 3791–3797.

[11] P. Vijayan, S. Subramanian, and B. Ravindran, "Multi-label collective classification in multi-attribute multi-relational network data," in IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining, 2014, pp. 509–514.

[12] C. Han, Q. Wu, M. K. Ng, J. Cao, M. Tan, and J. Chen, "Tensor based relations ranking for multi-relational collective classification," in Data Mining (ICDM), 2017 IEEE International Conference on. IEEE, 2017, pp. 901–906.

[13] H. Eldardiry and J. Neville, "Across-model collective ensemble classification," in AAAI, 2011.

[14] Y. Jacob, L. Denoyer, and P. Gallinari, "Classification and annotation in social corpora using multiple relations," in Proc. of the 20th ACM Int. Conf. on Information and Knowledge Management, 2011, pp. 1215–1220.

[15] D. Eswaran, S. Günnemann, C. Faloutsos, D. Makhija, and M. Kumar, "Zoobp: Belief propagation for heterogeneous networks," Proceedings of the VLDB Endowment, vol. 10, no. 5, 2017.

[16] M. Ji, J. Han, and M. Danilevsky, "Ranking-based classification of heterogeneous information networks," in Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011, pp. 1298–1306.

[17] Y.-R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher, "Metafac: community discovery via relational hypergraph factorization," in Proc. of the 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 2009, pp. 527–536.

[18] K. Maruhashi, F. Guo, and C. Faloutsos, "Multiaspectforensics: Pattern mining on large-scale heterogeneous networks with tensor analysis," in Advances in Social Networks Analysis and Mining (ASONAM), 2011 Int. Conf. on, 2011, pp. 203–210.

[19] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in Proc. of the 28th Int. Conf. on Machine Learning (ICML-11), 2011, pp. 809–816.

[20] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in Advances in Neural Information Processing Systems, 2013, pp. 2787–2795.

[21] R. Jenatton, N. L. Roux, A. Bordes, and G. R. Obozinski, "A latent factor model for highly multi-relational data," in Advances in Neural Information Processing Systems, 2012, pp. 3167–3175.

[22] M. K.-P. Ng, X. Li, and Y. Ye, "Multirank: co-ranking for objects and relations in multi-relational data," in Proc. of the 17th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 2011, pp. 1217–1225.

[23] X. Li, M. K. Ng, and Y. Ye, "Har: Hub, authority and relevance scores in multi-relational data for query search." in SDM, 2012, pp. 141–152.

[24] X. Li, M. K. Ng, and Y. Ye, "Multicomm: Finding community structure in multi-dimensional networks," IEEE Trans. on Knowledge and Data Engineering, vol. 26, no. 4, pp. 929–941, 2014.

[25] X. Li, Y. Ye, and M. K. Ng, "Multivcrank with applications to image retrieval," IEEE Trans. on Image Processing, vol. 25, no. 3, pp. 1396–1409, 2016.

[26] J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu, and Z. Chen, "Cubesvd: a novel approach to personalized web search," in Proc. of the 14th Int. Conf. on World Wide Web. ACM, 2005, pp. 382–390.

[27] T. H. Haveliwala, "Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search," IEEE Trans. on knowledge and Data Engineering, vol. 15, no. 4, pp. 784–796, 2003.

[28] H. Tong, C. Faloutsos, and J.-Y. Pan, "Random walk with restart: fast solutions and applications," Knowledge and Information Systems, vol. 14, no. 3, pp. 327–346, 2008.

[29] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[30] S. M. Ross, Introduction to probability models. Academic press, 2014.

[31] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. R. Salakhutdinov, "Neighbourhood components analysis," in Advances in neural information processing systems, 2005, pp. 513–520.

[32] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," Journal of Machine Learning Research, vol. 10, no. Feb, pp. 207–244, 2009.

[33] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in Proceedings of the 24th international conference on Machine learning. ACM, 2007, pp. 209–216.

[34] R. Kellogg, "Uniqueness in the schauder fixed point theorem," Proc. of the American Mathematical Society, vol. 60, no. 1, pp. 207–210, 1976.

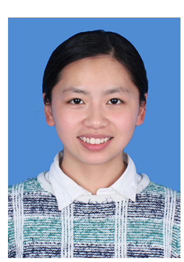
[35] M. Ji, Y. Sun, M. Danilevsky, J. Han, and J. Gao, "Graph regularized transductive classification on heterogeneous information networks," in Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2010, pp. 570–586.

[36] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "Nus-wide: a real-world web image database from national university of singapore," in Proceedings of the ACM international conference on image and video retrieval. ACM, 2009, p. 48.

[37] S. A. Macskassy, "Improving learning in networked data by combining explicit and mined links," in AAAI, vol. 22, 2007, pp. 590–595.

[38] Srivastava, Rupesh Kumar and Greff, Klaus and Schmidhuber, Jürgen, "Highway networks," in CoRR, 2015.

[39] Xiong Yun, Yizhou Zhang, Xiangnan Kong, Huidi Chen, and Yangyong Zhu, "GraphInception: Convolutional Neural Networks for Collective Classification in Heterogeneous Information Networks," in IEEE Transactions on Knowledge and Data Engineering, 2019.



Chao Han is currently a Post-Doctoral Researcher of the School of Electrical Engineering at Guangxi University, China. She received her Ph.D. degree in the School of Software Engineering from South China University of Technology, China, in 2019. She received her B.S. degree in Computer Science and Technology from Yunnan University, China, in 2013. Her research interests include data mining, machine learning, and graph learning.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3039533, IEEE Transactions on Knowledge and Data Engineering

Jian Chen is currently a Professor of the School of Software Engineering (SSE) at South China University of Technology, China. She joined SSE at South China University of Technology as a faculty member in 2005. She received her B.S. and Ph.D. degrees, both in Computer Science, from Sun Yat-Sen University, China, in 2000 and 2005 respectively. Her research interests can be summarized as developing effective and efficient data analysis techniques for complex data and the related applications. Particularly, she is currently interested in various techniques of data mining, Web search, information retrieval, recommendation techniques as well as their applications.

Mingkui Tan is currently working as a Professor with the School of Software Engineering at South China University of Technology. He received his Ph.D. degree in Computer Science from Nanyang Technological University, Singapore, in 2014. He received the Master degree in Control Science and Engineering in 2009 and his Bachelor degree in Environmental Science and Engineering in 2006, both from Hunan University in Changsha, China. His research interests include compressive sensing, big data learning, and large-scale optimization.

Michael K. Ng received the B.Sc. and M.Phil. degrees from the Universityof Hong Kong in 1990 and 1992, respectively, and the Ph.D. degree from the Chinese University of Hong Kong in 1995. He was a Research Fellow of Computer Sciences Laboratory at Australian National University from 1995 to 1997, and an Assistant/Associate Professor of the University of Hong Kong from 1997 to 2005. He was Professor/Chair Professor in Department of Mathematics at Hong Kong Baptist University from 2006 to 2019. He is currently a Chair Professor in Research Division of Mathematical and Statistical Science at The University of Hong Kong. His research interests include bioinformatics, image processing, scientific computing, and data mining. He is selected for the 2017 Class of Fellows of the Society for Industrial and Applied Mathematics. He obtained the Feng Kang Prize for his significant contributions in scientific computing. He serves on the Editorial Board members of several international journals.

Qingyao Wu is currently a Professor with the School of Software Engineering, South China University of Technology, China. He received the B.S. degree in software engineering from the South China University of Technology, and the M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology, China, in 2007, 2009, and 2013, respectively. He was a Post-Doctoral Research Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore, from 2014 to 2015. His current research interests include data mining, transfer learning, integrating vision and language.