Semi-supervised Learning over Heterogeneous Information Networks by Ensemble of Meta-graph Guided Random Walks

He Jiang[1] and Yangqiu Song[1] and Chenguang Wang[2] and Ming Zhang[3] and Yizhou Sun[4]
[1]Department of CSE, HKUST, Hong Kong. [2]IBM Research-Almaden, USA.
[3]School of EECS, Peking University, China. [4]Department of CS, UCLA, USA.
[1]hejiang@ust.hk, yqsong@cse.ust.hk, [2]chenguang.wang@ibm.com
[3]mzhang_cs@pku.edu.cn [4]yzsun@cs.ucla.edu

Abstract

Heterogeneous information network (HIN) is a general representation of many real world data. The difference between HIN and traditional homogeneous network is that the nodes and edges in HIN are with types. In many applications, we need to consider the types to make the decision more semantically meaningful. For annotation-expensive applications, a natural way is to consider semi-supervised learning over HIN. In this paper, we present a semi-supervised learning algorithm constrained by the types of HINs. We first decompose the original HIN schema into several semantically meaningful meta-graphs consisting of entity and relation types. Then random walk is performed to propagate the labels from labeled data to unlabeled data guided by the meta-graphs. After receiving labels from the results of random walk guided by meta-graphs, we carefully compare different ensemble algorithms to generate the final label with respect to all the clues from each meta-graphs. Experimental results on two knowledge based text classification datasets show that our algorithm outperforms traditional semi-supervised learning algorithms for HINs.

1 Introduction

Semi-supervised learning, a machine learning paradigm that learns from partially labeled data, has been well studied in machine learning community [Chapelle *et al.*, 2006]. One of the mainstream semi-supervised learning approaches is the so-called graph based semi-supervised learning [Zhu *et al.*, 2003; Zhou *et al.*, 2003]. Graph based semi-supervised learning views the data as a graph, e.g., manually constructed k-nearest-neighbor graph built on data similarities. Then it performs label propagation over the graph, which is regarded as a random walk with the labeled data being viewed as the "absorbing boundary" [Zhu *et al.*, 2003]. Zhou *et al.* (2003) further relaxed the random walk framework to be constrained learning by graph regularization. This framework corresponds to a generalized lazy random walk over the labeled graph [Zhou and Schölkopf, 2004], where the random walk considers an additional probability to stay at the current position.

In the real world, however, there are many kinds of data that can be naturally represented as heterogeneous information networks (HINs) [Sun and Han, 2012] rather than the homogeneous graph used by graph based semi-supervised learning. The difference between heterogeneous information networks and homogeneous networks is that the nodes and edges can be classified into different types. For example, social networks with users, tags, URLs, locations, etc., can be considered as an HIN [Kong *et al.*, 2013]. The scholar network, containing papers, authors, venues, keywords, is an HIN [Sun *et al.*, 2011]. The patient network, incorporated with gene network, drug network, and disease network, is also an HIN [Denny, 2012]. Moreover, the knowledge graphs, such as Freebase [Bollacker *et al.*, 2008] and Google Knowledge Vault [Dong *et al.*, 2014], are naturally HINs since all the entities and relations are typed with categories. When there are not enough annotations for certain types of nodes, semi-supervised learning can be considered. For example, we want to predict the users' genders in social network, classify the papers in scholar network into topics, group patients with potential diseases, and classify new entities based on the existing knowledge on knowledge graphs. Different classification problems still need a lot of labeling efforts. Thus, developing a semi-supervised learning algorithm over HINs can benefit a lot of real problems.

Semi-supervised learning over heterogeneous information network has a significant difference from original graph based semi-supervised learning, since the nodes and edges are with types. The labels propagating through different paths may have different effects. For example, if we consider a knowledge graph network with entity types *actor*, *director*, *movie*, *musician*, *singer*, and *song*, when we want to classify a specific entity, e.g., *Leonardo DiCaprio*, the labels that are propagated from other *actors* through *actors* and *directors* are more useful than the *actor* labels propagated through *singers* and *songs*. Thus, if we can have a strategy to guide the random walk over the heterogeneous information network, we can more effectively propagate the limited labels.

In this paper, we propose a meta-graph guided lazy random walk algorithm to guide the label propagation path with certain entity types. The meta-graph is an entity type network that characterizes the relationships between types, e.g., *actor*$\xrightarrow{\text{actIn}}$*movie*, *director*$\xrightarrow{\text{direct}}$*movie*, and *singer*$\xrightarrow{\text{sing}}$*song*, etc. When we constrain entity types of random walk, the random walk path follows two graphs: meta-graph and the original entity graph. We can enumerate a lot of meta-graphs based on the existing types of an HIN. Then after performing random walk guided by different meta-graphs, we ensemble the classification results using a supervised classifier, a maximum likelihood estimation of true labels given a lot of noisy labels [Dawid and Skene, 1979; Sheng *et al.*, 2008], as well as a co-training mechanism to jointly optimize the labels and the ensemble weights [Wan *et al.*, 2015]. We use knowledge graph (Freebase) enriched documents in 20-newsgroups and RCV1 datasets to demonstrate the effectiveness of semi-supervised learning over HIN, although other kinds of HINs should also be applicable. Extensive experiments show that by using HIN representation of documents, we can improve semi-supervised learning in a significant way. The code has been released at `https://github.com/HKUST-KnowComp/semihin`.

2 Related Work

In this section, we introduce the related work on semi-supervised learning on graphs or networks.

As we have described in the introduction, graph based semi-supervised learning has been well studied [Zhu *et al.*, 2003; Zhou *et al.*, 2003; Chapelle *et al.*, 2006]. In the context of graph link analysis in computer science community, the history of the research can be traced back to Pagerank [Page *et al.*, 1999] and HITS [Kleinberg, 1998] algorithms. When there are some annotation or preference on the nodes, personalized Pagerank can be used [Jeh and Widom, 2003; Haveliwala, 2003]. The formulation of personalized Pagerank is the same as Zhou's semi-supervised learning [Zhu *et al.*, 2003] although the meanings of the label/preference vectors are different. All the above algorithms assume that the graph has homogeneous type of nodes. The first work introduced heterogeneous information in random walk is used for recommendation problem [Brand, 2005], where the random walk is performed over a user-item bipartite graph.

For the recent development of HIN, there have been some attempts that use semi-supervised or side information to get better results for different tasks on HIN. For example, the entity similarities can be guided with partially labeled pairwise constraints [Sun *et al.*, 2012]. When documents can be represented as HINs using external knowledge graphs, pairwise constraints can also be used to guide document clustering [Wang *et al.*, 2015a]. Moreover, for a scholar network, transductive classification of entities on HIN has been developed [Ji *et al.*, 2010]. This algorithm discards the higher oder relationships but only uses the pairwise typed relations in the HIN. Recent study further extends this work by improving the weights on the network [Bangcharoensap *et al.*, 2016]. To avoid the single relation paths, topology shrinking sub-network algorithm [Wan *et al.*, 2015; Li *et al.*, 2016] is proposed to use meta-paths to first compute the similarities between the same type of nodes using a symmetric meta-path, and then it uses a linear combination of graph Laplacians computed from each similarity matrix as a whole to perform semi-supervised learning. Before performing semi-supervised learning, these methods need to compute the commuting matrices based on each meta-path, which is more costly than our approach. Moreover, there has been no existing work that attempts to use random walk over the original HIN for semi-supervised learning. The previously developed random walk process guided by meta-path [Lao and Cohen, 2010], however, can be a non-stationary process for some mata-paths for semi-supervised learning to converge.

Another line of research, which may not be called "semi-supervised learning over graph" but may be related is called collective classification [London and Getoor, 2014]. Collective classification uses the labeled nodes in the graph to predict unlabeled nodes. Different from pure random walk, collective classification assumes that the nodes can have features, e.g., the attributes of nodes, profiles of social users, etc. In the context of HIN, there has been some existing work using meta-paths to generate features for collective classification [Kong *et al.*, 2013].

3 Ensemble of Meta-graph Guided Random Walk Framework

In this section, we introduce the detailed algorithm of semi-supervised learning over HIN based on meta-graph guided random walk. We first analyze of lazy random walk over graph, and then show the key problems with meta-path and meta-graph guided random walk. Then we introduce different ensemble algorithms for multiple random walks.

3.1 Lazy Random Walk over Graph

Given a set of n nodes and corresponding edges, we can construct an adjacency matrix $\mathbf{W} \in \mathcal{R}^{n \times n}$. Then a lazy random walk over this graph considers a transition probability matrix:

$$\mathbf{P} = (1-\alpha)\mathbf{I} + \alpha\mathbf{W}\mathbf{D}^{-1}, \tag{1}$$

where $\mathbf{D}$ is the degree matrix with diagonal values $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$, and $\alpha \in (0,1)$ is a parameter controlling the probability of staying at the current position with probability $1-\alpha$ and moving to a random neighbor proportional to the weights on edge with probability α. There is an existing unique stationary distribution $\boldsymbol{\pi} \in \mathcal{R}^{n \times 1}$ satisfying

$$\boldsymbol{\pi} = \mathbf{P}\boldsymbol{\pi}. \tag{2}$$

We denote $T_{i,j} = \min\{t \geq 0 | V_t = v_j, V_0 = v_i, v_i \neq v_j\}$ as the first hitting time to node v_j starting from v_i, and denote $T_{i,i} = \min\{t > 0 | V_t = v_i, V_0 = v_i\}$ as the first returning time to v_j starting from v_i. The expectation of $T_{i,j}$ is the commonly used *hitting time*, which we denote as $\mathbf{H}_{ij}$. Then the commuting time between v_i and v_j is defined as $\mathbf{C}_{ij} = \mathbf{H}_{ij} + \mathbf{H}_{ji}$. Let $\mathbf{G} = (\mathbf{D} - \alpha\mathbf{W})^{\dagger}$ be the pseudo-inverse of $\mathbf{D} - \alpha\mathbf{W}$, then we have [Ham *et al.*, 2004]:

$$\begin{array}{ll} \mathbf{C}_{ij} \propto \mathbf{G}_{ii} + \mathbf{G}_{jj} - \mathbf{G}_{ij} - \mathbf{G}_{ji} & \text{if } v_i \neq v_j \\ \mathbf{C}_{ii} = 1/\boldsymbol{\pi}_i & \end{array}. \tag{3}$$

This relation is similar to the inner product similarity ($\mathbf{G}$) and norm distance ($\mathbf{C}$) in Euclidean space [Zhou and Schölkopf, 2004]. The longer the distance between v_i and v_j (larger $\mathbf{C}_{ij}$ commuting time), the smaller the similarity between v_i and v_j.

When there are labeled nodes in the graph, we perform random walk starting from the labeled data. To formulate the process, we denote a label vector for each class k as $\mathbf{l}_k \in \mathcal{R}^{n\times 1}$, where the labeled nodes are denoted as 1 while the other nodes as 0. Then the lazy random walk can be done by iteratively computing [Zhou *et al.*, 2003]:

$$\mathbf{f}_k^{t+1} = \alpha \mathbf{W}\mathbf{D}^{-1}\mathbf{f}_k^t + (1-\alpha)\mathbf{l}_k, \tag{4}$$

where $\mathbf{f}_k^{t+1}$ is the learned label vector for class k at time $t+1$. Note that this equation also often refers to personalized Pagerank when $\mathbf{l}_k$ characterizes nodes' preferences using some real values [Jeh and Widom, 2003].

The optimal value for $\mathbf{f}_k$ is:

$$\mathbf{f}_k = (\mathbf{I} - \alpha\mathbf{W}\mathbf{D}^{-1})^{\dagger}\mathbf{l}_k, \tag{5}$$

which corresponds to

$$\mathbf{f}_k(v_i) = \sum_{\mathbf{l}_k(v_j)=1} \bar{\mathbf{G}}_{ij}, \tag{6}$$

where $\bar{\mathbf{G}}_{ij} = \mathbf{G}_{ij}/\sqrt{\mathbf{C}_{ii}\mathbf{C}_{jj}}$ [Zhou and Schölkopf, 2004]. This means that the estimated label $\mathbf{f}_k(v_i)$ of v_i is the sum of $\bar{\mathbf{G}}_{ij}$ starting from labeled nodes. If v_i and v_j are more "similar" ($\bar{\mathbf{G}}_{ij}$ is greater), then the contribution of v_j as labeled data is greater for unlabeled data v_i. Then the assigned label by lazy random walk for unlabeled data is to choose the maximum $\mathbf{f}_k(v_j)$ from all $k = 1, \ldots, K$ classes for v_j.

3.2 Meta-path vs. Meta-graph Guidance

In this section, we discuss the difference between meta-path and meta-graph based random walk. Before going into the details, we briefly introduce the core concepts of heterogeneous information network [Sun *et al.*, 2011].

Definition 1 *A **heterogeneous information network** (HIN) is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an entity type mapping ϕ: $\mathcal{V} \rightarrow \mathcal{A}$ and a relation type mapping ψ: $\mathcal{E} \rightarrow \mathcal{R}$, where $\mathcal{V}$ denotes the entity set and $\mathcal{E}$ denotes the link set, $\mathcal{A}$ denotes the entity type set, and $\mathcal{R}$ denotes the relation type set.*

We can further use network schema to give a more abstractive description of the HIN.

Definition 2 *Given an HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the entity type mapping ϕ: $\mathcal{V} \rightarrow \mathcal{A}$ and the relation type mapping ψ: $\mathcal{E} \rightarrow \mathcal{R}$, the **network schema** for network G, denoted as $\mathcal{T}_{\mathcal{G}} = (\mathcal{A}, \mathcal{R})$, is a graph with nodes as entity types from $\mathcal{A}$ and edges as relation types from $\mathcal{R}$.*

One of the important concepts developed for HIN is the meta-path, the path defined over the entity types on the network schema [Sun *et al.*, 2011; Lao and Cohen, 2010].

Definition 3 *A **meta-path** $\mathcal{P}$ is a path defined on the graph of network schema $\mathcal{T}_G = (\mathcal{A}, \mathcal{R})$, and is denoted in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \ldots \xrightarrow{R_L} A_{L+1}$, which defines a composite relation $R = R_1 \cdot R_2 \cdot \ldots \cdot R_L$ between types A_1 and A_{L+1}, where $\cdot$ denotes relation composition operator, and L is the length of $\mathcal{P}$. A **commuting matrix** $\mathbf{M}_{\mathcal{P}}$ for a meta-path $\mathcal{P} = (A_1 - A_2 - \ldots - A_{L+1})$ is defined as $\mathbf{M}_{\mathcal{P}} = \mathbf{W}_{A_1A_2}\mathbf{W}_{A_2A_3}\ldots\mathbf{W}_{A_LA_{L+1}}$, where $\mathbf{W}_{A_iA_j}$ is the adjacency matrix between types A_i and A_j. $\mathbf{M}_{\mathcal{P}}(i,j)$ represents the number of path instances between objects x_i and y_j, where $\phi(x_i) = A_1$ and $\phi(y_j) = A_{L+1}$, under meta-path $\mathcal{P}$.*

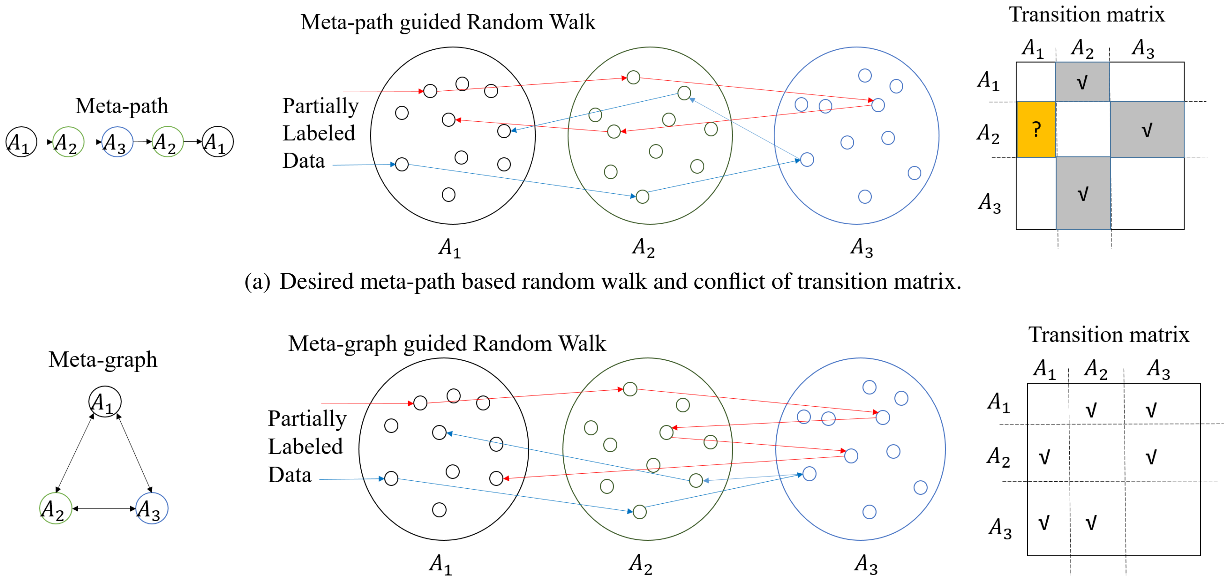
The difference between PathSim [Sun *et al.*, 2011] and path ranking algorithm (PRA) [Lao and Cohen, 2010] is PathSim normalizes the overall commuting matrix while PRA normalizes separate $\mathbf{W}_{A_iA_j}$'s.

Besides the meta-path, people have also found that meta-graph (or meta-structure) is very useful when defining the similarities between entities [Fang *et al.*, 2016; Huang *et al.*, 2016].

Definition 4 *A **meta-graph** $\mathcal{T}_s = (\mathcal{A}_s, \mathcal{R}_s)$ is a sub-graph of network schema $\mathcal{T}_G = (\mathcal{A}, \mathcal{R})$, where $\mathcal{A}_s \subseteq \mathcal{A}$ and $\mathcal{R}_s \subseteq \mathcal{R}$. We also denote the meta-graph derived subgraph of original HIN as $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$, where $\mathcal{V}_s \subseteq \mathcal{V}$ and $\mathcal{E}_s \subseteq \mathcal{E}$. The entities on the subgraph of HIN also follow the mapping ϕ: $\mathcal{V}_s \rightarrow \mathcal{A}_s$ and a relation type mapping ψ: $\mathcal{E}_s \rightarrow \mathcal{R}_s$.*

Now we show an example to illustrate why we work over meta-graphs rather than over meta-paths. Suppose we have an HIN with three entity types: A_1, A_2, and A_3. For example, we can think about *actor*, *director*, and *movie* with relations $\mathrm{marriage}$, actIn, actIn^{-1}, direct and direct^{-1}. One meta-path generated from the HIN is shown in the left in Figure 1(a). Suppose we have two labeled entities in type A_1. Then two typical paths of random walk starting from the labeled entities following the meta-path are shown in the middle in Figure 1(a). Note that a path of random walk following a meta-path should be constrained by the types in the meta-path [Lao and Cohen, 2010]. For example, the path $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5$ should follow $\phi(v_1) = A_1$, $\phi(v_2) = A_2$, $\phi(v_3) = A_3$, $\phi(v_4) = A_2$, and $\phi(v_5) = A_1$. Given this desired random walk, we try to formulate the transition matrix. From the right of Figure 1(a) that for $A_1 \rightarrow A_2 \rightarrow A_3 \rightarrow A_2$, we can easily fill in the sub-matrices based on $\mathbf{W}_{A_1,A_2}$, $\mathbf{W}_{A_2,A_3}$, $\mathbf{W}_{A_3,A_2}$. However, for the final random walk $A_2 \rightarrow A_1$ if we fill in with $\mathbf{W}_{A_2,A_1}$ and normalize the whole as probability distribution, then when we do random walk for $A_2 \rightarrow A_3$, there is also a probability to walk to entities with type A_1, which cannot strictly follow the meta-path. Thus, in this case, we need to either augment the meta-path to have an edge $A_2 \rightarrow A_1$ in parallel to $A_1 \rightarrow A_2$, or introduce another order of Markov chain to handle the type switching. For the former case, the meta-path is no longer a path, but rather a graph. For the letter case, a higher order random walk should be carefully designed depending on different meta-paths, where the storage of the stationary distribution should be handled carefully, e.g., by a non-Markovian random walk [Benson *et al.*, 2016].

To avoid the above problem with meta-path guided random walk, we propose to use meta-graph to guide the random walk. In the left of Figure 1(b), we show the meta-graph of fully connected bi-directional graph with nodes A_1, A_2, and A_3. Then in the middle of Figure 1(b), we show two typical random walk paths based on the constraints of meta-graph.



	A_1	A_2	A_3
A_1		√	
A_2	?		√
A_3		√	

(a) Desired meta-path based random walk and conflict of transition matrix.

	A_1	A_2	A_3
A_1		√	√
A_2	√		√
A_3	√	√	

(b) Meta-graph based random walk.

Figure 1: Comparison of meta-path based and meta-graph based random walks.

Finally in the right of Figure 1(b), we show the transition matrix of this random walk, which is consistent over time.

3.3 Ensemble

For a network schema, we can enumerate exponential number of meta-graphs. Thus, we should seek a better way to obtain sufficiently informative meta-graphs for us to use. One simple way is to first enumerate all the paths with certain lengths in the network schema. Then we automatically complete the meta-graph based on the selected meta-paths by checking the original network schema. Afterwards we have a set of meta-graphs that we can use for constraining the random walk. Here, we formally introduce the concept of meta-graph guided rand walk over HIN.

Definition 5 *A **meta-graph guided random walk over HIN** first obtains a set of meta-graphs $\mathcal{T}_{s_1}, \dots \mathcal{T}_{s_G}$ constructed from network schema $\mathcal{T}_G = (\mathcal{A}, \mathcal{R})$. Then we construct the corresponding adjacency matrices $\mathbf{W}^{(s_1)}, \dots, \mathbf{W}^{(s_G)}$ for the meta-graph derived subgraphs. For each $\mathbf{W}^{(s_i)}$ and for each class k, we run random walk for the estimated labels ${\mathbf{f}_k^{(s_i)}}^{t+1}$ in iteration t:*

$$ {\mathbf{f}_k^{(s_i)}}^{t+1} = \alpha \mathbf{W}^{(s_i)} {\mathbf{D}^{(s_i)}}^{-1} {\mathbf{f}_k^{(s_i)}}^{t} + (1-\alpha)\mathbf{l}_k \qquad (7) $$

where $\mathbf{D}^{(s_i)}$ is the degree matrix with diagonal values $\mathbf{D}_{ii}^{(s_i)} = \sum_j \mathbf{W}_{ij}^{(s_i)}$, and $\alpha \in (0,1)$ is a parameter controlling the probability of staying at the current position with probability $1-\alpha$ and moving to a random neighbor proportional to the weights on edge with probability α.

By running meta-graph guided random walk, we choose labels for the data by combining different estimated labels:

$$ \mathbf{f}_1^{(s_1)}(v_j), \dots, \mathbf{f}_k^{(s_1)}(v_j), \dots, \mathbf{f}_k^{(s_G)}(v_j), \dots, \mathbf{f}_K^{(s_G)}(v_j), \qquad (8) $$

where $\mathbf{f}_k^{(s_i)}(v_j)$ is the label of v_j generated by meta-graph s_i indicating whether it belongs to class k.

Given we have multiple label assignments from different random walks, we propose to use three meta-algorithms to find the final solution:

- SVM. Simply by exploiting the output scores of the meta-graph guided random walk, we use the labeled data to learn the linear combination of output scores. Given S_G meta-graphs and K classes, we learn a K-class Support Vector Machine (SVM) with $S_G \times K$ dimensional features.
- EM. We use the soft voting algorithm [Dawid and Skene, 1979] which can estimate the quality of each label vector $\hat{\mathbf{l}}^{(s_i)}$ (which is done by selecting the maximum value of $\hat{\mathbf{l}}^{(s_i)}(v_j) = \arg\max_k \mathbf{f}_k^{(s_i)}(v_j)$ from all $k = 1, \dots, K$ classes for v_j) to vote for the final label assignment for all the nodes we are interested in. Note that this voting algorithm has been improved in [Sheng *et al.*, 2008] for crowdsourcing with noisy labels, and in [Ipeirotis *et al.*, 2010] it shows that it can also incorporate partially labeled data.
- Co-training. Because each meta-graph carries different semantic information, each meta-graph is capable for classifying some samples and yields random results on other samples. Thus we use a co-training-like algorithm, to iteratively assign the soft labels for each meta-graph and the weight of the meta-graph for voting, which can propagate high confidence labels based on some meta-graphs to others. Our implementation is based on [Wan *et al.*, 2015].

4 Experiments

In this section, we present the results to show effectiveness and efficiency of our approach.

Table 1: Statistics of entities in different datasets: #(Doc) is the number of all documents; similar for #(Word) (# of words), #(FBEntity) (# of Freebase entities), and #Type (the total # of entity types).

	#(Doc)	#(Word)	#(FBEntity)	#(Type)
20NG-SIM	3,000	8,010	11,192	219
20NG-DIF	3,000	9,182	13,297	251
GCAT-SIM	3,596	11,096	10,540	233
GCAT-DIF	2,700	13,291	13,179	261

4.1 Datasets

We use two datasets to evaluate different algorithms.

20Newsgroups (20NG): The 20newsgroups dataset [Lang, 1995] contains about 20,000 newsgroups documents evenly distributed across 20 newsgroups.[1]

RCV1: The RCV1 dataset is a dataset containing manually labeled newswire stories from Reuter Ltd [Lewis *et al.*, 2004]. The news documents are categorized with respect to three controlled vocabularies: industries, topics and regions. There are 103 categories including all nodes except for root in the topic hierarchy. We select top category GCAT (Government/Social) to perform classification. In total, we have 60,608 documents with 16 leaf categories.

For both datasets, we obtained the semantic parsing results based on [Wang *et al.*, 2015a] which are now publicly available[2]. We follow [Wang *et al.*, 2016] to use four subsets of these to datasets to test our algorithms, which are **20NG-SIM** (comp.graphics, comp.sys.mac.hardware, and comp.os.ms-windows.misc), **20NG-DIF** (rec.autos, comp.os.mswindows.misc, and sci.space), **GCAT-SIM** (GWEA (Weather), GDIS (Disasters), and GENV (Environment and Natural World)), and **GCAT-DIF** (GENT (Arts, Culture, and Entertainment), GODD (Human Interest), and GDEF (Defense)). The statistics of the four dataset are summarized in Table 1. After meta-path selection [Wang *et al.*, 2015b] and further pruning low-frequency entities, we use nine augmented meta-graphs for 20NG datasets and eight meta-graphs for GCAT datasets based on the meta-paths.

4.2 Baseline Methods

We test the performance of our semi-supervised classification algorithm with different groups of baseline methods.

First, we test different types of features for text classification. Our algorithm is general for HINs. However, here we use knowledge augmented graph as HIN for text classification. Thus, a natural baseline for us is to see whether the knowledge we add in should be represented as HIN instead of other features. Here we compare two types of features for traditional machine learning algorithms:

BOW: Traditional bag-of-words model with term frequency weighting mechanism.

BOW+Entity: BOW augmented with additional features from entities in grounded knowledge from Freebase. This setting incorporates knowledge as flat features.

Second, we test different graph based semi-supervised learning mechanisms:

LP: We use LP to denote the traditional graph based label propagation algorithm operated based on similarity graph constructed by data dependent features [Zhou *et al.*, 2003]. We empirically select 10-nearest-neighbors for all the experiments.

LP-Meta-path: We implemented a simplified version of the state-of-the-art meta-path based semi-supervised learning algorithm [Wan *et al.*, 2015]. It uses meta-paths to first compute PathSims [Sun *et al.*, 2011] and the corresponding Laplacians. Then it jointly learns the propagated labels and the weights for different meta-paths to propagate the labels.

LP-KnowSim: We also implemented a simplified version of KnowSim [Wang *et al.*, 2015b], an unsupervised meta-path weighting based similarity, for the meta-paths we used to generate the similarities between documents and then construct the 10-nearest-neighbor based graph for label propagation.

SemiHIN-DWD: This is the simple bipartite graph version of our algorithm, which only considers the document-word relationships. In this case, our algorithm reduces to semi-supervised learning on bipartite graphs.

SemiHIN-Full-Graph: We also compare our algorithm with random walk over the full parsed graph. In our ensemble, we also incorporate this full graph.

SemiHIN-Ensemble: As shown in Section 3.3, we proposed three ways of ensemble of different predictions. Here, to simplify notations, we denote them as **Ensemble-SVM**, **Ensemble-EM**, and **Ensemble-Co-train**.

All the semi-supervised learning is performed with the same fixed controlling parameter $\alpha = 0.98$ shown in Eq. (4). To make sure having the best performance, before performing random walk, we also performed unsupervised feature selection [He *et al.*, 2006] for SemiHIN-Full-Graph, and applied the selection weights to each feature when computing ensemble of random walks.

4.3 Comparison

We first show the comparison results for all the configurations in Table 2. All the experiments are trained with five labeled data for each class. We evaluate algorithms in a transductive setting, which means we check whether an algorithm can use the five labels for each class to classify all the remaining examples correctly. Thus, all the results are averaged numbers of classification accuracy with 50 random trials. For supervised learning algorithms such as naive Bayes (NB) and support vector machine (SVM), they can only see the labeled data. For all semi-supervised learning algorithms, they can see the whole data including both labeled and unlabeled data.

From Table 2 we can see that, supervised learning with BOW+entity is comparable and often better than BOW, since we add more features about entities. LP based semi-supervised learning on 20NG datasets is better than supervised learning with the same amount of labeled data, since it leverages the unlabeled data. However, for GCAT datasets, LP based semi-supervised learning is slightly worse than supervised learning. This may be because for GCAT data, there are some words very class-indicative so that when

[1] http://qwone.com/~jason/20Newsgroups/

[2] https://github.com/cgraywang/TextHINData

Table 2: Performance of different classification algorithms on 20NG-SIM, 20NG-DIF, GCAT-SIM, and GCAT-DIF datasets. We show our results of five labeled training data for each class. All the numbers are averaged accuracy (in percentage %) over 50 random trials.

	NB		SVM		LP			SemiHIN		Ensemble		
Settings Datasets	BOW	BOW+ Entity	BOW	BOW+ Entity	BOW+ Entity	Meta-path	Know-Sim	DWD Graph	Full-Graph	SVM	EM	Co-train
20NG-SIM	39.02	48.46	37.34	49.67	54.53	57.75	56.87	48.94	58.46	52.04	54.44	**60.99**
20NG-DIF	43.74	57.24	39.57	55.71	72.40	76.13	77.14	61.31	77.69	71.36	73.08	**80.08**
GCAT-SIM	71.24	71.24	73.92	74.64	70.97	71.05	60.59	79.14	**81.02**	68.79	69.96	80.97
GCAT-DIF	56.60	56.66	63.52	63.91	61.95	61.37	51.64	64.32	65.05	57.48	58.19	**66.95**

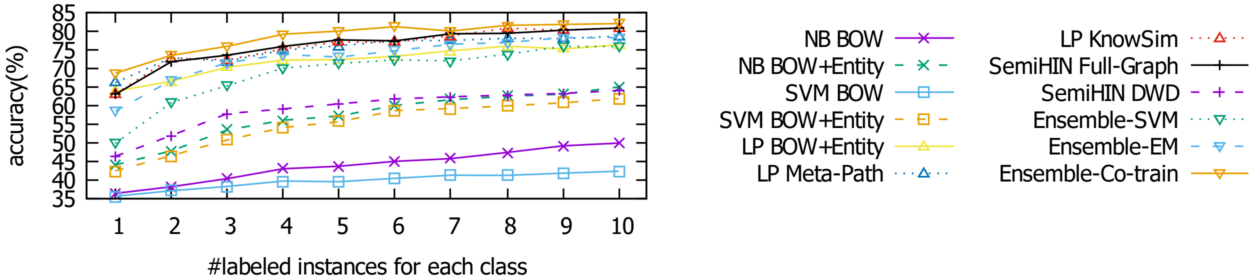


Figure 2: Classification results on 20NG-DIF dataset with different numbers of labeled documents per class.

converting i.i.d. features to similarities (which are used in LP), it introduces more ambiguity.

For HIN based algorithms, we found LP-Meta-path is better than LP based on BOW+Entity features. This makes sense since LP-Meta-path also incorporates DWD path, which is based on BOW. Co-training seems effective to mutually boost different meta-paths. LP-Meta-path is also comparable or better than KnowSim, since KnowSim is only unsupervised ensemble of meta-path based similarities while LP-Meta-path co-trains the weights of different meta-paths.

SemiHIN-Full-Graph is better than LP-BOW+Entity. This means the structural information among entities indeed helps improving semi-supervised learning. For the ensemble results, in general SVM is the worst, EM is better, and Co-train performs best. Ensemble-Co-train is better than SemiHIN-Full-Graph on 20NG data and GCAT-DIF. This may be because there are some other meta-graphs (or meta-paths) producing better results than the full graph. Then co-training can bootstrap the final labeling accuracy. Ensemble can help us automatically find a good solution without trying different meta-graphs based on the limited partially labeled data. Comparing Ensemble-Co-train and LP-Meta-path, it is shown that Ensemble-Co-train is better. An additional benefit of not working with PathSim is that we do not need to compute PathSim which could be more computational costly in practice.

Besides the overall results, we show results on 20NG-DIF dataset with different numbers of labeled data for each class in Figure 2. From the figure we can see that, with more labels, all the algorithms' classification results can be improved. In general, all the results are consistent with Table 2.

4.4 Computational Time

The computation of random walk with the sparse transition matrix is at worst $O(N^3)$. As we observed that the undirected graph Laplacian is semi-positive definite, we can replace the inversion with a conjugate gradient descent (CGD) algorithm. For sparse matrix, the CGD method can achieve $O(m\sqrt{r})$ time, where m is the number of links, and r represents the condition number of the sparse matrix. We report the time based on a retail laptop with an Intel i7-4750HQ CPU and 16 Gigabytes RAM. For 20NG-SIM data, SemiHIN with matrix multiplication and inversion costed $52,410$ seconds while it only costed 1.8 seconds with CGD. Both NB and SVM costed less than 1 second. The original LP costed 2.2 seconds.

5 Conclusion

In this paper, we present a meta-graph guided random walk ensemble algorithm over heterogeneous information networks for semi-supervised learning. We first propose the undirected meta-graph structure and apply a graph-based semi-supervised learning algorithm. Then we combined predictions from different meta-graphs using three different ensemble algorithms. We demonstrated that our approach outperforms other state-of-the-art traditional and HIN based semi-supervised learning algorithms. We believe meta-graph is a general representation of many graphs. We would also study different graphs using meta-graph in the future.

Acknowledgements

This paper was supported by HKUST WeChat WhatLab, China 973 Fundamental R&D Program (No.2014CB340304), National Natural Science Foundation of China (NSFC Grant Nos. 61472006 and 91646202), and NSF Career Award #1741634.

References

[Bangcharoensap *et al.*, 2016] Phiradet Bangcharoensap, Tsuyoshi Murata, Hayato Kobayashi, and Nobuyuki Shimizu. Transductive classification on heterogeneous information networks with edge betweenness-based normalization. In *WSDM*, pages 437–446, 2016.

[Benson *et al.*, 2016] Austin R. Benson, David F. Gleich, and Lek-Heng Lim. The spacey random walk: a stochastic process for higher-order data. *CoRR*, abs/1602.02102, 2016.

[Bollacker *et al.*, 2008] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250, 2008.

[Brand, 2005] Matthew Brand. A random walks perspective on maximizing satisfaction and profit. In *SDM*, pages 12–19, 2005.

[Chapelle *et al.*, 2006] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006.

[Dawid and Skene, 1979] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[Denny, 2012] Joshua C. Denny. Chapter 13: Mining electronic health records in the genomics era. *PLoS Computational Biology*, 8(12), 2012.

[Dong *et al.*, 2014] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *KDD*, pages 601–610, 2014.

[Fang *et al.*, 2016] Yuan Fang, Wenqing Lin, Vincent Wenchen Zheng, Min Wu, Kevin Chen-Chuan Chang, and Xiaoli Li. Semantic proximity search on graphs with metagraph-based learning. In *ICDE*, pages 277–288, 2016.

[Ham *et al.*, 2004] Jihun Ham, Daniel D. Lee, Sebastian Mika, and Bernhard Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *ICML*, 2004.

[Haveliwala, 2003] Taher H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Trans. Knowl. Data Eng.*, 15(4):784–796, 2003.

[He *et al.*, 2006] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In *NIPS*, pages 507–514. 2006.

[Huang *et al.*, 2016] Zhipeng Huang, Yudian Zheng, Reynold Cheng, Yizhou Sun, Nikos Mamoulis, and Xiang Li. Meta structure: Computing relevance in large heterogeneous information networks. In *SIGKDD*, pages 1595–1604, 2016.

[Ipeirotis *et al.*, 2010] Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *KDD Workshop on Human Computation*, pages 64–67, 2010.

[Jeh and Widom, 2003] Glen Jeh and Jennifer Widom. Scaling personalized web search. In *WWW*, pages 271–279, 2003.

[Ji *et al.*, 2010] Ming Ji, Yizhou Sun, Marina Danilevsky, Jiawei Han, and Jing Gao. Graph regularized transductive classification on heterogeneous information networks. In *ECML/PKDD*, pages 570–586, 2010.

[Kleinberg, 1998] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In *SODA*, pages 668–677, 1998.

[Kong *et al.*, 2013] Xiangnan Kong, Jiawei Zhang, and Philip S. Yu. Inferring anchor links across multiple heterogeneous social networks. In *CIKM*, pages 179–188, 2013.

[Lang, 1995] Ken Lang. Newsweeder: Learning to filter netnews. In *ICML*, pages 331–339, 1995.

[Lao and Cohen, 2010] Ni Lao and William W. Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 81(1):53–67, 2010.

[Lewis *et al.*, 2004] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *JMLR*, 5:361–397, 2004.

[Li *et al.*, 2016] Xiang Li, Ben Kao, Yudian Zheng, and Zhipeng Huang. On transductive classification in heterogeneous information networks. In *CIKM*, pages 811–820, 2016.

[London and Getoor, 2014] Ben London and Lise Getoor. Collective classification of network data. In *Data Classification: Algorithms and Applications*, pages 399–416. 2014.

[Page *et al.*, 1999] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, Nov. 1999.

[Sheng *et al.*, 2008] Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *KDD*, pages 614–622, 2008.

[Sun and Han, 2012] Yizhou Sun and Jiawei Han. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3(2):1–159, 2012.

[Sun *et al.*, 2011] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *PVLDB*, pages 992–1003, 2011.

[Sun *et al.*, 2012] Yizhou Sun, Brandon Norick, Jiawei Han, Xifeng Yan, Philip S. Yu, and Xiao Yu. Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. In *KDD*, pages 1348–1356, 2012.

[Wan *et al.*, 2015] Mengting Wan, Yunbo Ouyang, Lance M. Kaplan, and Jiawei Han. Graph regularized meta-path based transductive regression in heterogeneous information network. In *SDM*, pages 918–926, 2015.

[Wang *et al.*, 2015a] Chenguang Wang, Yangqiu Song, Ahmed El-Kishky, Dan Roth, Ming Zhang, and Jiawei Han. Incorporating world knowledge to document clustering via heterogeneous information networks. In *KDD*, pages 1215–1224, 2015.

[Wang *et al.*, 2015b] Chenguang Wang, Yangqiu Song, Haoran Li, Ming Zhang, and Jiawei Han. Knowsim: A document similarity measure on structured heterogeneous information networks. In *ICDM*, pages 1015–1020, 2015.

[Wang *et al.*, 2016] Chenguang Wang, Yangqiu Song, Haoran Li, Ming Zhang, and Jiawei Han. Text classification with heterogeneous information network kernels. In *AAAI*, pages 2130–2136, 2016.

[Zhou and Schölkopf, 2004] Dengyong Zhou and Bernhard Schölkopf. Learning from labeled and unlabeled data using random walks. In *Pattern Recognition, 26th DAGM Symposium, August 30 - September 1, 2004, Tübingen, Germany, Proceedings*, pages 237–244, 2004.

[Zhou *et al.*, 2003] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NIPS*, pages 321–328, 2003.

[Zhu *et al.*, 2003] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.