A Relation Prediction Method Based on PU Learning

Gao-Jing Peng, Ke-Jia Chen, Shijun Xue, Bin Liu
Jiangsu Key Laboratory of Big Data Security & Intelligent Processing
Nanjing University of Posts and Telecommunications
Nanjing, Jiangsu 210046, China
penggj_njupt@163.com, chenkj@njupt.edu.cn, xueshijun_2010@163.com, bins@ieee.org

***Abstract*—This paper studies relation prediction in heterogeneous information networks under PU learning context. One of the challenges of this problem is the imbalance of data number between the positive set *P* (the set of node pairs with the target relation) and the unlabeled set *U* (the set of node pairs without the target relation). We propose a *K*-means and voting mechanism based technique *SemiPUclus* to extract the reliable negative set *RN* from *U* under a new relation prediction framework *PURP*. The experimental results show that PURP achieves better performance than comparative methods in DBLP co-authorship network data.**

Keywords-link prediction; relation prediction; heterogeneous information networks; PU learning

I. INTRODUCTION

Link prediction aims to predict the formation possibility of missing links or future links in a network based on the network's current or historical data. It has a wide range of applications, such as citation prediction in a bibliographic dataset, product recommendation in an e-commerce service, online advertisement click prediction in an online network and so on [1]. Most of the existing link prediction methods are proposed for homogeneous information networks where there is only one single type of nodes and edges.

However, types of nodes and edges in real networks are usually multiple. These networks are called heterogeneous information networks (HINs). In HIN, structural dependencies of different relations also increase the difficulty of link prediction [2] [3] [4]. Recently, Sun et al. [5] used the concept of meta-path in HINs and proposed relation prediction problem, which can be seen as an extension of link prediction problem. Here is an example of relation prediction in a co-authorship network (Figure 1). The network includes four types of nodes and ten types of links. The target relation to predict is the co-authorship between any author pair, which can be represented by the meta-path $Author \xrightarrow{write} Paper \xrightarrow{write^{-1}} Author$.

Relation prediction can be treated as a supervised learning process. If the target relation exists between a_1 and a_2, the label of node pair $<a_1, a_2>$ is set to "+1", otherwise it is set to "-1". This process normally requires a lot of positive examples and negative examples to train the model. However, the number of negative examples is often limited or not available in many real-world fields. The PU learning technique will enable the use of positive and unlabeled examples to construct a classification model.

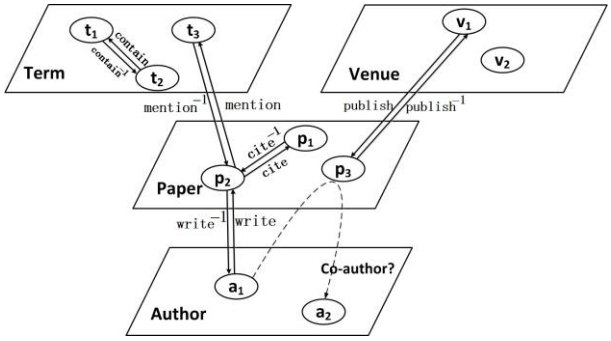

Figure 1. The co-authorship network

In the above co-authorship network, if the target relation between a_1 and a_2 does not exist at the moment, it does not mean that the target relation will not form in the future. So the label of a node pair without the target relation is better set to "0" instead of "-1". With this assumption, all node pairs are now divided into the positive example set *P* and the unlabeled example set *U*.

PU learning has become a new research topic in the field of classification. Though widely used in text mining, graph mining and so on [7] [8] [9], it was not used in link mining until recent years. In 2014, Zhang et al. [6] used PU learning for the first time to predict anchor links between multiple networks. They used the Spy technique to extract reliable negative examples. Different from their work, this paper aims to predict the target relation and does not limit to links in a single HIN.

The main challenges of relation prediction in HIN are:

- *Extraction of reliable negative examples*
  The most important challenge of PU learning is to extract reliable negative examples *RN* from *U*. But for relation prediction in HIN, most of the existing semi-supervised PU learning methods are not suitable any more. It is necessary to design a new efficient method.
- *Heterogeneity of network*
  The types of nodes or links are multiple in HIN, so traditional link prediction methods of homogeneous information networks are no longer applicable. Also, dependencies of relations between nodes and heterogeneity of links bring great difficulties for the prediction task.
- *Link sparsity*

978-1-5386-1829-5/17/$31.00 ©2017 IEEE

In a network, the number of possible links approaches $O(n^2)$ (given n nodes). When n is very large, the network will be very sparse and the number of P and U will be extremely unbalanced. This problem is aggravated in HINs since the prediction target could be a relation (combined links) between different types of nodes. How to use the large amount of unlabeled data on the network remains to be a challenge.

In this paper, we propose a new relation prediction framework PURP to address the above challenges. First, PURP uses meta-path (below a fixed path length) to define all the relations between two nodes and extract the topological structure based on meta-path for each node pair. Secondly, a new negative extraction technique SemiPUclus is proposed, which is based on *K*-means and voting mechanism. Finally, a classifier is trained using the set composed of *P* and *RN*.

The rest of this paper is organized as follows. Section II gives the formulation and definition of the problem. In Section III, the details of SemiPUclus and PURP are introduced. The experimental results are presented and analyzed in Section IV. Section V discusses some related works and Section VI concludes the paper.

II. PROBLEM FORMULATION

Definition 1. **Heterogeneous Information Network:** A heterogeneous information network can be represented as a directed graph $G = (V_1 \cup V_2 \ldots \cup V_k, E_1 \cup E_2 \ldots \cup E_h)$. Where $V_1 \cup V_2 \ldots \cup V_k$ represents the set of nodes with k types, $E_1 \cup E_2 \ldots \cup E_h$ represents the set of links with h types.

Figure 2 shows the network schema of a bibliographic information network DBLP. In DBLP, authors can collaborate with each other to write papers. A paper may contain certain terms and can be published in a journal or conference. Thus, DBLP can be formulated as a directed graph $G = (V, E)$, where $V = V_P \cup V_C \cup V_A \cup V_T$ and $E = E_{P,P} \cup E_{P,C} \cup E_{P,A} \cup E_{P,T} \cup E_{A,A}$, here P (*Paper*), C (*Conference*), A (*Author*) and T (*Term*) represent types of nodes. Note that all links in E have its reverse link. For example, there are two types of links "*write*" and "*write*$^{-1}$" in the link set $E_{P,A}$, which represent the relation that an author writes papers and that papers are written by the author.

Definition 2. **Meta-path [5]:** Given a network schema $T_G = (N_1 \cup N_2 \ldots \cup N_K, L_1 \cup L_2 \ldots \cup L_h)$, where $N_x (x = 1, 2, \ldots, k)$ denotes the type of node set V_x and $L_y (y = 1, 2, \ldots, h)$ denotes the type of link set E_y. A meta-path can be represented as the form of $N_1 \xrightarrow{L_1} N_2 \xrightarrow{L_2} N_3 \xrightarrow{L_3} \ldots N_{n-1} \xrightarrow{L_{n-1}} N_n$. Thus, a relation type $R = L_1 \circ L_2 \circ \ldots L_{n-2} \circ L_{n-1}$ is defined between node types N_1 and N_n, where $\circ$ represents a composite operation on link types.

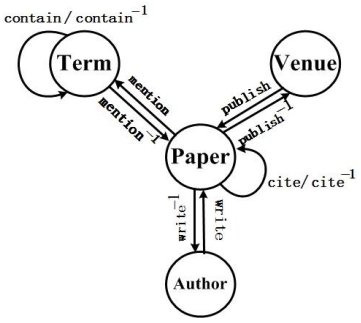

Figure 2. The network schema of DBLP

Definition 3. **Inverse Path [5]:** Given a meta-path $\rho = N_1 \xrightarrow{L_1} N_2 \xrightarrow{L_2} N_3 \xrightarrow{L_3} \ldots N_{n-1} \xrightarrow{L_{n-1}} N_n$, ρ^{-1} expressed as $N_n \xrightarrow{L_{n-1}^{-1}} N_{n-1} \xrightarrow{L_{n-2}^{-1}} N_{n-2} \xrightarrow{L_{n-3}^{-1}} \ldots N_2 \xrightarrow{L_1^{-1}} N_1$ is the inverse path of ρ.

Note that each meta-path in this paper has the same type of starting and ending nodes. For example, the meta-path $Author \xrightarrow{write} Paper \xrightarrow{write^{-1}} Author$ has the same type "*Author*" at the starting and ending nodes. All meta-paths used in this paper are shown in Figure 3.

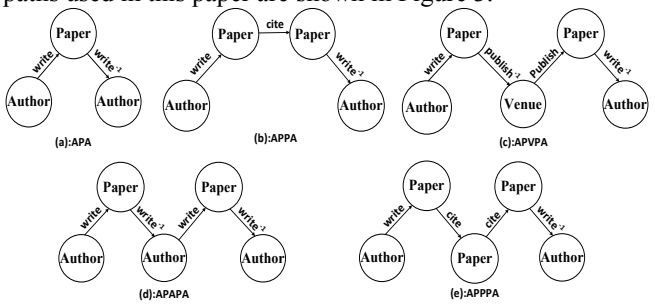

Figure 3. Meta-paths in DBLP

Problem 1. **Link Prediction in HIN:** Given a directed graph $G = (V, E)$ following the specific network schema $T_G = (N, L)$, the task of link prediction is to study whether a target link E_x between node x_i and x_j will exist or not.

Problem 2. **Relation Prediction in HIN:** Different from link prediction in HIN, the task of relation prediction in HIN is to study whether a target relation between node x_i and x_j will exist or not.

As the prediction problem in HIN is not limited to the $|L|$ types of links in the $T_G = (N, L)$, the prediction target can be the composite of existing types of links, which can be defined as relation prediction problem instead. For example, if we want to predict the co-authorship between the two authors in the DBLP network, while there is no direct "*Co-author*" links in the DBLP schema, we can use the composite of "*write*" and "*write*$^{-1}$" links to express "*Co-author*" relation. This paper represents the "*Co-author*" relation by using the meta-path $Author \xrightarrow{write} Paper \xrightarrow{write^{-1}} Author$, abbreviated as "APA".

Definition 4. **Positive and Unlabeled Examples in Prediction Problem:** Given a node pair $< x_i, x_j >$, if a target relation (in relation prediction problem) or a target link (in link prediction problem) exists between node x_i and x_j, node pair $< x_i, x_j >$ is labeled positive. Otherwise, it is an unlabeled example.

For example, Given a DBLP network $G = (V_P \cup V_C \cup V_A \cup V_T, E_{P,P} \cup E_{P,C} \cup E_{P,A} \cup E_{P,T} \cup E_{A,A})$, each author pair $< x_i, x_j >$ with the co-authorship "APA" belongs to the positive set *P*; otherwise, the unlabeled set *U*. So $P = E_{A,A}$, $U = V_A \times V_A - P$.

Problem 3. **Relation Prediction based on PU learning:** Given *P* and *U*, the key task of relation prediction based on PU learning is to find the reliable negative set *RN* from *U* by a certain negative extraction technique. Then, a binary classifier will be trained using *P* and *RN* under supervised learning settings. This classifier can predict whether the target relation between node x_i and x_j will exist or not.

III. THE PROPOSED RELATION PREDICTION FRAMEWORK

This section introduces the details of the proposed PU learning framework PURP to solve relation prediction in HIN.

A. Heterogeneous Topological Features

First of all, the representation of each example in PURP should be discussed. The topological features reflect the connection properties between node pairs in the network. Many commonly used topological features have been proposed in homogeneous networks [12]. However, most of these features are no more meaningful in HIN since the node types and link types in HIN are diverse. In this paper, we also use the topological features based on meta-path proposed by Sun et al. [2]. Given the meta-path set *MP* and node pair $< x_{t,i}, x_{t,j} >$, where node $x_{t,i}$ refers to the type of node x_i is *t*. Each meta-path ρ between the starting node $x_{t,i}$ and the ending node $x_{t,j}$ can be quantified by measures of *path count* and *random walk*.

- *Path count*
  Path count measures the number of path instances between the starting node $x_{t,i}$ and the ending node $x_{t,j}$ following a given meta-path ρ, defined as $PC_\rho(< x_{t,i}, x_{t,j} >)$.
- *Random walk*
  Random walk is defined in (1), where $PC_\rho(< x_{t,i}, \cdot >)$ denotes the total number of paths following the meta-path ρ starting with node $x_{t,i}$.

$$PW_\rho = (< x_{t,i}, x_{t,j} >) = \frac{PC_\rho(< x_{t,i}, x_{t,j} >)}{PC_\rho(< x_{t,i}, \cdot >)} \quad (1)$$

All topological features mentioned above focus on topological structure of a network. Moreover, we use the label information in the training set to calculate the weights of all topological features for each meta-path, which can be seen as the approximation problem (2).

$$\zeta(< x_{t,i}, x_{t,j} >) = Y(< x_{t,i}, x_{t,j} >) - \sum_{k=1}^{|MP|} \sum_{h=1}^{m} w_{k,h} \times f_{k,h}(< x_{t,i}, x_{t,j} >) \quad (2)$$

$$W^{opt} = \arg\min_W \| \sum_{i=1}^{n} \sum_{j=1}^{n} (\zeta(< x_{t,i}, x_{t,j} >)) \| \quad (3)$$

In equation (3), *n* represents the number of target nodes, and $f_{k,h}(< x_{t,i}, x_{t,j} >)$ is the h^{th} topological feature of the target node based on the k^{th} meta-path ρ. Given a meta-path, the matrix *Y* is constructed by the connection state between the target node pairs, and matrix *Y* as follows:

$$Y(< x_{t,i}, x_{t,j} >) = \begin{cases} 1, <x_{t,i}, x_{t,j}> with\ \rho \\ 0, \quad otherwise \end{cases} \quad (4)$$

The gradient descent method [22] can be used to get the weight of each topological feature. Then the *H* highest weight topological features are selected as the final feature set.

B. SemiPUclus Algorithm

In this section, a new *K*-means and voting mechanism based algorithm SemiPUclus (Algorithm 1) is proposed. This algorithm aims to find the reliable negative set *RN* efficiently.

Algorithm 1 SemiPUclus

Input: *N* local positive clusters LP_i, the unlabeled set *U* and the number *K*
Output: Reliable negative set *RN*
Variable Description: *RN* is a set of reliable negative examples and *LN* is a set of likely negative examples.

1. $RN = \varnothing$, $LN = \varnothing$;
2. Get *K* local unlabeled clusters $ULC_i\ (i = 1, 2, \ldots, K)$ by *K*-means method;
3. **for** i=1 to *N* **do**
   for j=1 to *K* **do**
   Compute the Euclidean distance $d(ULC_j, LP_i)$ between ULC_j and LP_i;
   Add $d(ULC_j, LP_i)$ into set D_i;
   end for
   Sort the distances in D_i in decreasing order;
   Find the median distance $DMedian_i$ in D_i;
   end for
4. **for** i=1 to *N* **do**
   for each distance $d(ULC_j, LP_i) \in D_i$ **do**
   if $d(ULC_j, LP_i) > DMedian_i$ **do**
   add the corresponding ULC_j into set LN_i;
   end if
   end for
   end for
5. **for** i=1 to *N* **do**
   for j=1 to *K* **do**
   if ULC_j occurs in LN_i **do**
   $Count_j$++;
   end if
   end for

end for

6. Find the cluster ULC_j with the highest value of $Count_j$ as $MULC$;
7. $RN = RN \cup MULC$;
8. Return RN;
9. **end**

The core idea of the algorithm is to use the voting mechanism that minority obeys majority. Each local positive cluster $LP_i(1, 2, ..., n)$ votes the likely negative clusters in the K local unlabeled clusters, and then the likely negative clusters with the largest number of votes in short $MULC$s are considered as the reliable negative set RN. It should be noted that there may exist more than one $MULC$, since the likely negative clusters may have the same number of votes.

As a simple example shown in Figure 4, local positive clusters LP_1 and LP_2 respectively vote the local unlabeled clusters ULC_2, ULC_5, ULC_7, ULC_8 and ULC_3, ULC_6, ULC_7, ULC_8, which are considered as the likely negative clusters. Local clusters ULC_7 and ULC_8 with the largest votes are considered as the reliable negative set RN.

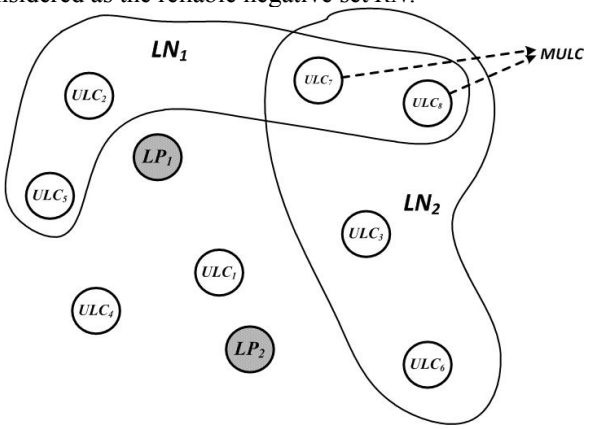

Figure 4. Extraction RN from the unlabeled set U

C. PURP Framework

In PURP framework, all reliable negative examples are first selected by SemiPUclus algorithm, a classifier C_F is then constructed by using RN and P as the train set. The details of the PURP framework are as follows:

Prediction Framework PURP

Input: a heterogeneous information network $G = (V, E)$, the meta-path set MP and the target node set T.

1. Get the label matrix Y between each target node pair in set $T \times T$ by the equation (4);
2. For each meta-path $\rho \in MP$ and node pairs compute the m topological features in HIN;
3. The weights of topological features are computed by equation (2);
4. Select the top H features with the largest weight;
5. Generate the positive set P and the set U with selected features from node pairs in $T \times T$ and the label matrix Y;
6. Find the reliable negative set RN by SemiPUclus algorithm;
7. Use an existing supervised learning algorithm to train the set P and RN and get the final prediction result.

IV. EXPERIMENTS

A. Data Preparation

In this paper, the DBLP dataset is used to evaluate the performance of the proposed method and other comparative methods, which can be downloaded from http://dblp.uni-trier.de/xml/. The schema of the DBLP network is shown in Figure 2, and the experiments use the data between 2008 and 2013, which contain 546,777 users, 437,613 papers, 32,974 venues and 1,276,113 author-writes-paper links. The meta-paths used in experiments are shown in Figure 3. They represent possible relations between authors. Considering the scale and efficiency of the experiment, the length of each meta-path is no more than 4.

B. Comparison Methods

The following three comparison PU learning methods also use relation "APA" as the prediction target. The features are calculated and selected by the same set of meta-paths (detailed in Section 3) used in PURP.

- *PU-simple*
  The PU-*simple* method is always used in the traditional binary link prediction. The method regards author pairs with relation "APA" as positive examples labeled as "+1", otherwise as negative examples labeled as "-1".
- *PU-spy* [6]
  The PU-*spy* method is proposed for the anchor link prediction between HINs. The method constructs two classification models in order to extract the reliable negative examples from U. As shown in Figure 5, the PU-*spy* method first randomly selects 15% examples from the set P with label "+1" to form the "*spy*" set SP. There are two training processes. In the first training process, the original labels "+1" of the instances in set SP are hidden and the instances in set $SP \cup U$ are labeled as "0". The minimal formation probability ε can be obtained by predicting the label of examples in SP. ε is then considered as the threshold value in the next testing process. In the second training process, the original set U is viewed as the negative set N that all instances are labeled "-1". Finally, the reliable negative set RN forms, where the formation probability of each example is less than ε.

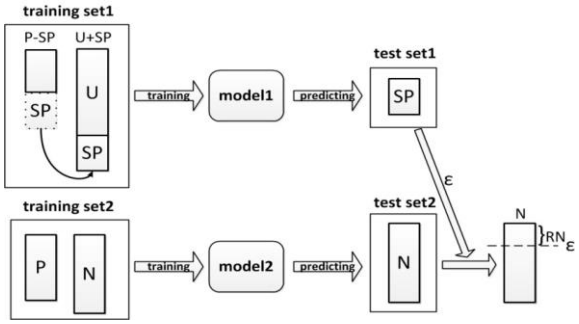

Figure 5. Spy technique in PU-*spy*

- *PU-NB* [17]
  The PU-*NB* method is proposed for text classification, we apply it to link prediction and relation prediction for the first time. The method also constructs two classification models to extract the reliable negative examples from *U*. Firstly, the set *P* with label "+1" and the set *U* with label "-1" are used to establish prediction model M_N using Naive Bayes classifier. Secondly, the examples in *U* whose labels are predicted by M_N as "-1" are viewed as the reliable negative examples.

After extracting reliable negative set *RN* from *U*, the above three methods use Naive Bayes classifier to establish the final prediction models.

C. Experiment Setups

All the above methods including the proposed methods are coded by Java. The supervised learning algorithm used in PURP framework and the *K*-means clustering algorithm used in SemiPUclus are implemented based on WEKA API. The lab machine is a laptop computer with a CPU of Intel (R) Core (TM) I3 2310M CPU @ 2.10GHZ, memory of 6GB.

The DBLP data are divided into two datasets D_1 and D_2 according to the publication year $T_0 = [2008,\ 2010]$ and $T_1 = [2011,\ 2013]$. D_1 is the initial training set containing 826,438 node pairs and D_2 is the testing set containing 354,186 node pairs. Note that node pairs with co-authorship (meta-path "APA") in training phrase are viewed as positive examples, others are viewed as unlabeled examples. Node pairs who have the target relation in testing phrase but not in training phrase are supposed to be predicted as positive examples and those who have no target relation in the entire dataset $D_1 \cup D_2$ are supposed to be predicted as negative examples.

D. Experiment Results

In this section, the experiments results of all relation prediction methods based on PU learning are compared: PU-*simple*, PU-*NB* [17], PU-*spy* [6] and PURP. The metrics of AUC [6] (Area under ROC) and Running Time are used to evaluate the performance of all the methods.

The experiment was repeated for 10 times according to the different ratios of the positive examples. The α stands for the ratio of the positive examples in training set. For example, $\alpha = 0.05$ means that the number of positive examples is only 5% of the training dataset and the rest examples are unlabeled examples.

Figure 6 shows the AUC comparison of four methods with different ratios of positive examples in DBLP. The figure seems that PURP outperforms all comparison methods in all settings. For example, when the value of α is 10%, the AUC value of PURP is significantly higher than PU-*spy* and PU-*NB*. For PU-*spy* method, when the value of α is changed from 35% to 40%, the AUC value drops sharply. The AUC values of PURP and PU-*simple* remain stable under different values of α.

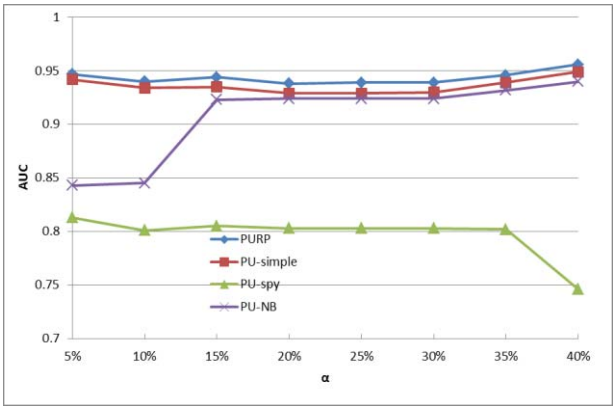

Figure 6. Performance comparison of four methods in DBLP

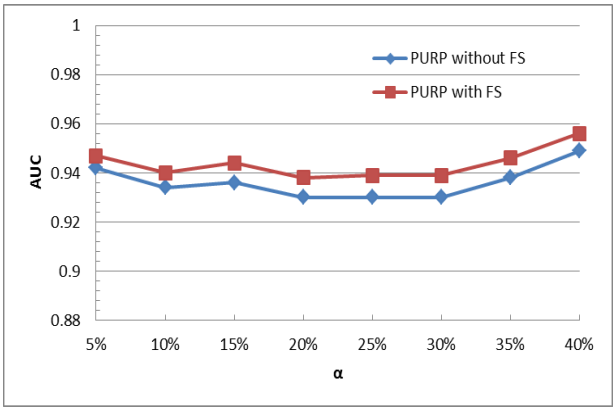

Figure 7. Performance comparison of PURP with FS and PURP without FS

To verify the effectiveness of feature selection (FS) on PURP, we compare the performance of PURP with FS and PURP without FS. The results are shown in Figure 7. The PURP with feature selection performs better in almost all settings of α (5%, 10%, 15%, … , 40%). For example, when the value of α is 5% or 10%, the AUC value of PURP with FS is slightly higher than PURP without FS. However, when the value of α is 25% or 30%, the AUC value improved is close to 0.01. This proves that not all topological features are helpful to improve the performance of predictors.

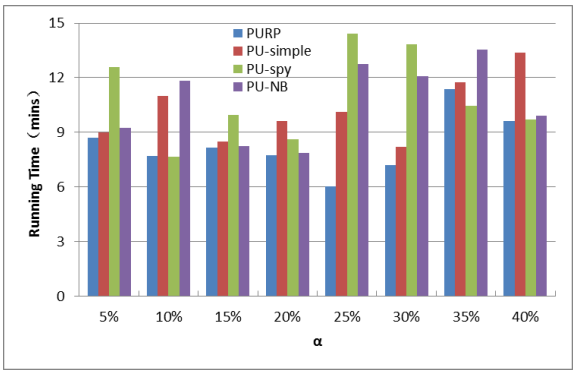

Figure 8. Running time of different methods

The running time of all the methods are compared on the same hardware configuration (Figure 8.). Note that the running time includes training time and testing time. In most settings of α, PURP (with FS) requires shortest time. The overall time complexity of PURP is $O(|MP|\times n^2)+O(F_{opt})+O(K\times N\times d\ (ULC_j, LP_i))+2\times O_{cluster}$. Where $O(|MP|\times n^2)$ is the time complexity of calculating topological features, $O(F_{opt})$ is the time complexity of optimization, and $O(K\times N\times d\ (ULC_j, LP_i))+2\times O_{cluster}$ represents the time complexity of SemiPUclus. Considering that all methods we compared calculate topological features in the same way. We ignore the time complexity $O(|MP|\times n^2)$ in calculation of running time.

V. Related Works

This section introduces related works on link prediction in HIN and several representative PU learning methods.

A. Link Prediction in HIN

In recent years, link prediction in HIN has become a hot topic. Researchers have proposed many link prediction methods in different backgrounds and for different tasks.

Sun et al. [5] [21] proposed the concept of "*meta-path*" to improve the graph distance metric and extended link prediction problems HIN into the relation prediction problems in HIN. Davis et al. [13] used the triad combination information in HIN to assist link prediction. Essentially, this method uses heterogeneous features under the supervised learning framework.

Some unsupervised methods [1] [14] studied prediction of edges with a single type independently. For example, Kuo et al. [1] proposed an unsupervised probabilistic graph model using aggregation statistics, which can be used to predict the unknown types without training data in HIN. Subsequently, researchers shift their focus to the studies of multi-union heterogeneous online social networks. Kong et al. [15] proposed the concept of "multi-union heterogeneous social network" and "anchor link" to predict anchor links which are across networks. Zhang et al. [4] [16] studied the problems of collective link prediction across partially union networks. They proposed to transfer information from existing mature networks to union networks in order to help new users predict links with multiple types.

Our method is mainly used for relation prediction but can certainly be used for link prediction as well. The main innovation of our method is the use of PU learning technology, which is still rare in the context as far as we know.

B. PU Learning

In the supervised learning framework, the traditional methods need a lot of positive examples and negative examples to train the model. In fact, the number of negative examples in many real-world fields is often limited or not available, which facilitates the studies of only using positive and unlabeled examples to construct a classification model. That is the PU learning problem.

PU learning technology has been widely used in various fields such as text classification and graph mining. For example, Liu et al. [7] [8] [17] proposed many different methods to extract reliable negative examples from the unlabeled set in the field of text mining. Zhao et al. [9] used PU learning in the field of graph mining. In 2014, Zhang et al. [6] proposed the concept of "connection state" and "formation state" and use spy technology to extract reliable negative examples in the field of link prediction. But the purpose of their work is to predict links between multiple networks. Our method is to predict target *relations* only for a single HIN rather than a multi-union HIN. Most of the above methods follow a two-phase strategy: the first step is to extract *RN* from *U*; the second step is to use *P* and *RN* to train a binary classifier.

In some methods [18] [19] [20], PU learning is applied to time series data. The number of instances in *P* is too small to represent the feature spaces. Different from the two-phase strategy above, Ratanamahatana et al. [18] studied to extend *P* iteratively using an early stop standard. Nguyen et al. [20] proposed to use the cluster chains and decision boundaries to extend *P* and *RN*.

Our method uses *K*-means based reliable negative examples extraction technique. When data scale is large, it is easy to shorten the calculation time and mitigate data imbalance between *P* and *U*.

VI. Conclusion

In this paper, we study the problem on training an effective relation predictor by using a large number of unlabeled instances in HIN. A new negative example extraction technique SemiPUclus using *K*-means based voting mechanism and a new relation prediction framework PURP are proposed. Experiments show that the performance of relation prediction can be significantly improved in PURP framework.

However, the main advantages of our method are not limited to this. Different from traditional supervised learning methods, our methods uses the information of unlabeled set so that it can achieve automatic labeling to reliable negative examples. Moreover, our method is suitable for predicting target relations that can be encoded by any topological features not limited to meta-paths in HIN.

In future work, we will try to use other means except meta-paths to represent relation in the network. The proposed method will also be experimented in multi-network to predict relations between different types of nodes.

ACKNOWLEDGMENT

This research was supported by the National Natural Science Foundation of China (No.61571238 and No.61603197).

REFERENCES

[1] T. T. Kuo, R. Yan, Y. Y. Huang, P. H. Kung, and S. D. Lin, "Unsupervised link prediction using aggregative statistics on heterogeneous social networks," Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013, pp. 775-783, doi: 10.1145/2487575.2487614.

[2] Y. Sun, J. Han, Y. Yang, and N. V. Chawla, "Link prediction in heterogeneous networks: Influence and time matters," Proceedings of the IEEE International Conference on Data Mining, 2012.

[3] J. B. Lee and H. Adorna, "Link Prediction in a modified heterogeneous bibliographic network," Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2012, pp. 442-449, doi: 10.1109/ASONAM.2012.78.

[4] J. Zhang, X. Kong and P. S. Yu, "Predicting social links for new users across aligned heterogeneous social networks," Proceedings of the IEEE International Conference on Data Mining, Dallas, TX, 2013, pp. 1289-1294, doi: 10.1109/ICDM.2013.134.

[5] Y. Sun, J. Han, X. Yan, P. Yu and T. Wu, "Pathsim: meta path-based top-k similarity search in heterogeneous information networks," Proceedings of the VLDB Endowment, 2011, pp. 992-1003.

[6] J. Zhang, P. S. Yu and Z.-H Zhou, "Meta-path based multi-network collective link prediction," Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 1286-1295, doi: 10.1145/2623330.2623645.

[7] X. Li, B. Liu, and S. K. Ng, "Learning to identify unexpected instances in the test set," Proceedings of IJCAI, 2007, pp. 2802-2807.

[8] X. L. Li, B. Liu and S. K. Ng, "Negative training data can be harmful to text classification," Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2010, pp. 218-228.

[9] Y. Zhao, X. Kong and P. S. Yu, "Positive and Unlabeled Learning for Graph Classification," Proceedings of the IEEE International Conference on Data Mining, 2011, pp. 962-971, doi: 10.1109/ICDM.2011.119.

[10] L. M. Manevitz and M. Yousef, "One-class SVMs for document classification," Journal of Machine Learning Research, 2001, pp. 139-154.

[11] B. Liu, Y. Dai, X. Li, W. S. Lee and P. S. Yu, "Building text classifiers using positive and unlabeled examples," Proceedings of the IEEE International Conference on Data Mining, 2003, pp. 179-186, doi: 10.1109/ICDM.2003.1250918.

[12] D. Liben‐Nowell and J. Kleinberg, "The link prediction problem for social networks," Journal of the Association for Information Science and Technology, Vol. 58, 2007, pp. 1019-1031, doi: 10.1002/asi.20591.

[13] D. Davis, R. Lichtenwalter and N. V. Chawla, "Multi-relational link prediction in heterogeneous information networks," Proceedings of International Conference on Advances in Social Networks Analysis and Mining, 2011, pp. 281-288, doi: 10.1109/ASONAM.2011.107.

[14] L. Eronen and H. Toivonen, "Biomine: predicting links between biological entities using network models of heterogeneous databases," BMC Bioinformatics, Vol. 13, 2012, pp. 89-95, doi: 10.1186/1471-2105-13-119.

[15] X. Kong, J. Zhang and P. S. Yu, "Inferring anchor links across multiple heterogeneous social networks," Proceedings of the ACM International Conference on Information & Knowledge Management 2013, pp. 179-188, doi: 10.1145/2505515.2505531.

[16] J. Zhang, X. Kong and P. Yu, "Transferring heterogeneous links across location-based social networks," Proceedings of the ACM International Conference on Web Search and Data Mining, 2014, pp. 303-312, doi: 10.1145/2556195.2559894.

[17] B. Liu, Y. Dai, X. Li, W. S. Lee and P. S. Yu, "Building text classifiers using positive and unlabeled examples," Proceedings of the IEEE International Conference on Data Mining, 2003, pp. 179-186, doi: 10.1109/ICDM.2003.1250918.

[18] C. A. Ratanamahatana and D. Wanichsan, "Stopping criterion selection for efficient semi-supervised time series classification," Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Vol. 149, 2008, pp. 1-14.

[19] L. Wei and E. Keogh, "Semi-supervised time series classification," Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pp. 748-753.

[20] M. N. Nguyen, X. L. Li and S. K. Ng, "Positive unlabeled leaning for time series classification," Proceedings of IJCAI, Vol. 11, 2011, pp.1421-1426.

[21] Y. Sun, J. Han, C. C. Aggarwal and N. V. Chawla, "When will it happen?: relationship prediction in heterogeneous information networks," Proceedings of the ACM international Conference on Web Search and Data Mining, 2012, pp. 663-672.

[22] D. C. Montgomery, E. A. Peck and G. G. Vining, "Introduction to linear regression analysis," John Wiley & Sons, 2015.