On Transductive Classification in Heterogeneous Information Networks

Xiang Li, Ben Kao, Yudian Zheng, Zhipeng Huang
The University of Hong Kong, Pokfulam Road, Hong Kong
{xli2, kao, ydzheng2, zphuang}@cs.hku.hk

ABSTRACT

A heterogeneous information network (HIN) is used to model objects of different types and their relationships. Objects are often associated with properties such as labels. In many applications, such as curated knowledge bases for which object labels are manually given, only a small fraction of the objects are labeled. Studies have shown that transductive classification is an effective way to classify and to deduce labels of objects, and a number of transductive classifiers have been put forward to classify objects in an HIN. We study the performance of a few representative transductive classification algorithms on HINs. We identify two fundamental properties, namely, *cohesiveness* and *connectedness*, of an HIN that greatly influence the effectiveness of transductive classifiers. We define metrics that measure the two properties. Through experiments, we show that the two properties serve as very effective indicators that predict the accuracy of transductive classifiers. Based on cohesiveness and connectedness we derive (1) a black-box tester that evaluates whether transductive classifiers should be applied for a given classification task and (2) an active learning algorithm that identifies the objects in an HIN whose labels should be sought in order to improve classification accuracy.

Keywords

heterogeneous information network; transductive classification; knowledge base

1. INTRODUCTION

Networks (or graphs) model real world entities and their relationships by objects and links (or edges). A heterogeneous information network (HIN) is a network whose objects are of different types and whose links represent different kinds of relationships between objects. Compared with homogeneous information networks (in which all objects/links are of one single type), an HIN is much more expressive in capturing complex real-world entities and their relationships. HINs are used in many data sources. These HINs vary in terms of their complexities — from relatively simple bibliographic networks (e.g., DBLP) to very complex knowledge bases. A representative example of the latter is Yago[1], which captures information derived from Wikipedia, WordNet and GeoNames. Yago is a repository of information on more than 10 million entities (such as persons, organizations, cities, etc.) and it records more than 120 million facts about these entities. Another example is Freebase[2], which is a community-curated knowledge base of well-known people, places and things. Information on Yago and Freebase can be modeled as RDF graphs, which are examples of HINs.

To enrich the information of HINs, objects are often associated with labels. For example, authors in DBLP can be labeled by their areas of research and movies in IMDB can be labeled by their genres. These descriptive labels facilitate information retrieval and knowledge understanding, and they allow interesting logical deductions on the data to be made. Labeling objects in HINs, however, requires very costly manual efforts. For large HINs, such as Yago and Freebase, we observe that only a small fraction of the objects are given their desired labels. For example, we inspected movie objects on Yago. We found that around 75% of the adventure movies are not properly labeled with the genre. The problem of missing labels severely limits the knowledge bases in their support of knowledge reasoning.

In recent years, a number of classification algorithms have been devised to deduce object labels in an HIN. Generally, classification methods can be categorized into inductive classification and transductive classification. Inductive methods [7, 16, 11, 15] use objects with known labels to train a model with which the labels of unknown objects are derived. Transductive methods [9, 10, 24, 26], on the other hand, utilize the "relatedness" between objects to "propagate" labels. For example, if a labeled object x is connected by an edge to an unlabeled object y, then the label of x is propagated to y because the two objects are related by an edge relation. Besides edge relations, objects can also be related by "path relations", which, in the context of HINs, are often called *meta-paths.* A meta-path is a schematic sequence of object types. For example, consider DBLP. If A and P represent object types *author* and *paper*, respectively, and that an edge between an object of type A and an object of type P represents *authorship*, then the meta-path A-P-A expresses the *co-authorship* relation between author objects. How "strongly" the label of an object x influences (or

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA
© 2016 ACM. ISBN 978-1-4503-4073-1/16/10. . . $15.00
DOI: http://dx.doi.org/10.1145/2983323.2983730

[1] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago

[2] https://www.freebase.com/

Table 1: Accuracies of transductive classifiers

Dataset	% of labeled objects	GNetMine	HetPathMine	Grempt
DBLP	0.5%	88.0%	86.1%	89.3%
Yago	5%	47.5%	48.4%	49.2%
Freebase	5%	63.7%	64.7%	65.4%

is propagated to) another object y depends on the strength of the relations between the two objects. We will elaborate more on transductive classifiers in Section 3. Typically, inductive classification requires a substantial set of training (labeled) data to construct an accurate model. For HINs with scarce labeled data, transductive methods are more effective. Hence, we focus on transductive classifiers.

We studied existing transductive classifiers on HINs [4, 8, 18] and made two observations: (1) We performed a *cross-sectional* study (applying the algorithms on the same HIN classification tasks) and found that given the same task, the accuracies of the classifiers are comparable. (2) We performed a *longitudinal* study (applying the same algorithm across different HIN tasks) and found that the performance of a transductive classifier varies greatly over different tasks.

To illustrate, we apply three HIN transductive classifiers, namely, GNetMine [4], HetPathMine [8] and Grempt [18] on three HINs: DBLP, Yago and Freebase. (We will describe the three classifiers in detail in Section 3.) For DBLP, the task is to classify authors into their research areas. For Yago and Freebase, the tasks are to classify movie objects into their genres. (More details on these datasets and classification tasks will be given in Section 4.) Table 1 shows part of the experimental results. If we look at a cross section (a row) of the table, we see that the three algorithms give very similar accuracies for the same task. For example, the accuracies range from 47.5% to 49.2% in the classification of Yago movies. On the other hand, if we look at the table longitudinally (along a column), the performance of each algorithm varies greatly. For example, GNetMine is 47.5% accurate in classifying Yago movies but it is 88% accurate in classifying DBLP authors. Noting that for DBLP, in our experiment, only 0.5% of the author objects are labeled (i.e., are included in the training set), which is 10 times smaller than those of the other tasks, the differences in accuracy across the classification tasks are very drastic. From our observations, we argue that in the study of transductive classifiers on HINs, it is perhaps not necessary to spend much efforts in fine tuning the classification algorithms, as that would only bring marginal benefits. Rather, one should analyze the intrinsic properties of an HIN and the classification task so as to understand the factors that impact the success of transductive classification. Our objective is to shed light on the latent principles behind transductive classification in HINs and to provide insightful reference for further research on the topic. Our main contributions are summarized as follows.

- While previous works focus mostly on the design of classification algorithms, we give an in-depth analysis on data (HINs) and classification tasks. We identify two influential factors, namely *cohesiveness* and *connectedness*, that generally affect the effectiveness of transductive classifiers. Intuitively, an HIN is highly cohesive if object relations are mostly between objects of the same label and an HIN is highly connected if objects of the same label are all or mostly "related". As we have mentioned above, relatedness between two objects refers to how well they are connected via edges and paths in an HIN. Hence, cohesiveness and connectedness are structural properties of a network for a given classification task. For HINs of low cohesiveness and connectedness, we show that transductive classification performs poorly regardless of the algorithm used. We propose quantitative measures of cohesiveness and connectedness and discuss how these measures can be practically estimated given an HIN.

Table 2: Descriptions of symbols

Notation	**Description**
$G = (V, E)$	An HIN G with object set V and link set E
$\mathcal{T}, T_i, m$	$\mathcal{T} = \{T_1, T_2, ..., T_m\}$, a set of m object types
$\mathcal{X}_i, \mathcal{X}_i^{\mathcal{L}}, \mathcal{X}_i^*$	The set of (␣/labeled/unlabeled) type T_i objects
$T_G = (\mathcal{T}, \mathcal{R})$	Network schema of HIN G
$p_{x_u \rightsquigarrow x_v} \vdash \mathcal{P}$	$p_{x_u \rightsquigarrow x_v}$ is an instance of the meta-path $\mathcal{P}$
$\mathcal{L} = \{l_1, ..., l_k\}$	A set of k labels
$G_{T_i, \mathcal{P}}$, $G_{\mathcal{P}}$	TSSN of object type T_i induced by meta-path $\mathcal{P}$
L, $\mathcal{C}_L$	A labeling, a label-induced clustering
Υ, Ψ	Cohesiveness and connectedness

- We design a *black-box tester* that evaluates an HIN and a classification task. The tester estimates the HIN's cohesiveness and connectedness and recommends whether transductive classification should be applied. We carry out case studies and show that the tester is accurate in making its recommendation.
- We propose an active learning strategy ALCC (Active Learning based on Cohesiveness and Connectedness). Active learning is about wisely selecting objects for which labels are sought. A good active learning strategy would select those objects that improve the classification accuracy the most. ALCC aims at identifying those objects that bring the best improvement in cohesiveness and connectedness, leading to good improvement in classification accuracy. We show that ALCC compares favorably against other active learning methods.

The rest of the paper is organized as follows. Section 2 gives some basic definitions. Section 3 mentions related works. In Section 4, we analyze HINs and classification tasks. We give formal definitions of cohesiveness and connectedness, and present measures that quantify the two properties. In Section 5, we discuss how the two properties can be leveraged in deriving a black-box tester and an active learner. Section 6 presents experimental results. Finally, Section 7 concludes the paper.

2. DEFINITIONS

In this section we give some basic definitions. Table 2 summarizes some of the symbols used in this paper.

Definition 1. **Heterogeneous Information Network (HIN)** [4]. Let $\mathcal{T} = \{T_1, ..., T_m\}$ be a set of m object types. For each type T_i, let n_i and $\mathcal{X}_i = \{x_{i1}, ..., x_{in_i}\}$ be the number and the set of objects of type T_i, respectively. An HIN is a graph $G = (V, E)$, where $V = \bigcup_{i=1}^{m} \mathcal{X}_i$, and E is a set of links, each represents a binary relation between two objects in V. If $m = 1$ (i.e., there is only one object type), G reduces to a homogeneous information network. □

Definition 2. **Network schema** [13]. A network schema is the meta template of an HIN $G = (V, E)$. Let (1) $\phi : V \rightarrow \mathcal{T}$ be an object-type mapping that maps an object in V into its type, and (2) $\psi : E \rightarrow \mathcal{R}$ be a link-relation mapping that maps a link in E into a relation in a set of

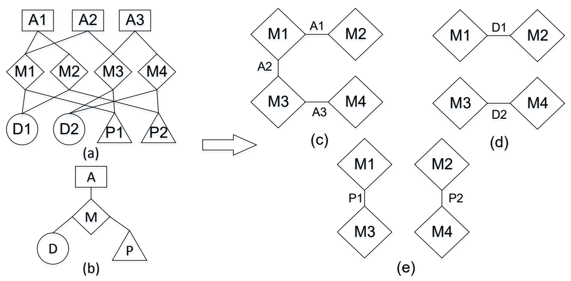


Figure 1: An HIN (a), its schematic graph (b), TSSNs derived from meta-paths MAM (c), MDM (d), and MPM (e)

relations $\mathcal{R}$. The network schema of an HIN G, denoted by $T_G = (\mathcal{T}, \mathcal{R})$, shows how objects of different types are related by the relations in $\mathcal{R}$. T_G can be represented by a *schematic graph* with $\mathcal{T}$ and $\mathcal{R}$ being the node set and the edge set, respectively. Specifically, there is an edge (T_i, T_j) in the schematic graph iff there is a relation in $\mathcal{R}$ that relates objects of type T_i to objects of type T_j. □

Figure 1(a) shows an example HIN that models movie information. The HIN consists of four object types: $\mathcal{T}$ = { movie (◇), actor(□), director(○), producer(△) }. There are also three relations in $\mathcal{R}$, which are illustrated by the three edges in the schematic graph (Figure 1(b)). For example, the relation between *actor* and *movie* carries the information of which actor has acted in which movie.

Definition 3. **Meta-path** [13]. A meta-path $\mathcal{P}$ is a path defined on the schematic graph of a network schema. A meta-path $\mathcal{P}$: $T_1 \xrightarrow{R_1} \cdots \xrightarrow{R_l} T_{l+1}$ defines a composite relation $R = R_1 \circ \cdots \circ R_l$ that relates objects of type T_1 to objects of type T_{l+1}. If two objects x_u and x_v are related by the composite relation R, then there is a path, denoted by $p_{x_u \rightsquigarrow x_v}$, that connects x_u to x_v in G. Moreover, the sequence of links in $p_{x_u \rightsquigarrow x_v}$ matches the sequence of relations R_1, ..., R_l based on the link-relation mapping ψ. We say that $p_{x_u \rightsquigarrow x_v}$ is a *path instance* of $\mathcal{P}$, denoted by $p_{x_u \rightsquigarrow x_v} \vdash \mathcal{P}$. □

As an example, the path $p_{M1 \rightsquigarrow M3} = M1 \rightarrow A2 \rightarrow M3$ in Figure 1(a) is an instance of the meta-path Movie-Actor-Movie (abbrev. MAM). Meta-paths are used in many data mining tasks on HINs. Grempt, for example, utilizes meta-paths that relate objects of the same type (i.e., $T_1 = T_{l+1}$) to obtain homogeneous sub-networks through which labels are propagated to perform transductive classification.

Definition 4. **Topology Shrinking Sub-network (TSSN)** [18]. Given an HIN $G = (V, E)$, the TSSN of a certain object type T_i derived from a meta-path $\mathcal{P}$ is a graph whose nodes consist of only objects of type T_i and whose edges connect objects that are related by instances of $\mathcal{P}$. Formally, the TSSN is the graph $G_{T_i,\mathcal{P}} = (\mathcal{X}_i, E_{T_i})$, where $E_{T_i} = \{e_{uv} | p_{x_u \rightsquigarrow x_v} \vdash \mathcal{P}, x_u, x_v \in \mathcal{X}_i\}$. We write $G_{\mathcal{P}}$ instead of $G_{T_i,\mathcal{P}}$ if the object type T_i is implicitly known. □

Figures 1(c)-(e) show the TSSNs of type Movie derived from the meta-paths MAM, MDM and MPM, respectively. A TSSN shows how objects of a certain type are related by the composite relation given by a meta-path. For example, the meta-path MAM relates two movie objects if those movies share an actor. In Figure 1(c), movies M1 and M2 are connected by an edge in the TSSN because actor A1 acted in both movies.

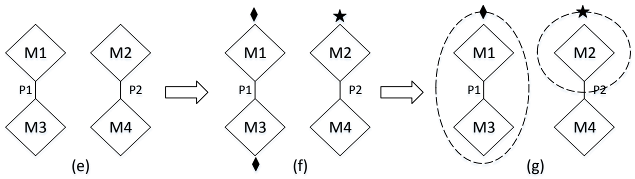


Figure 2: Labeling and label-induced clustering

Definition 5. **Labeling**. Given an HIN $G = (V, E)$ and a set of labels $\mathcal{L}$ for objects of type T_i, a labeling is a mapping $L : \mathcal{X}_i \rightarrow \mathcal{L} \cup \{*\}$, where "$*$" denotes missing label. For any $x \in \mathcal{X}_i$, if $L(x) = *$, we say that x is *unlabeled*; otherwise, $L(x)$ is called the *label* of x. We use $\mathcal{X}_i^{\mathcal{L}}$ and $\mathcal{X}_i^{*}$ to denote the set of labeled objects and the set of unlabeled objects in $\mathcal{X}_i$, respectively. □

Definition 6. **Label-induced clustering**. Given an HIN $G = (V, E)$, a set of labels $\mathcal{L} = \{l_1, ..., l_k\}$ for objects of type T_i, and a labeling L, the label-induced clustering $\mathcal{C}_L = \{C_1, ..., C_k\}$ is a partitioning of the set $\mathcal{X}_i^{\mathcal{L}}$ into k clusters such that C_j contains all and only those objects whose labels are l_j (i.e., $C_j = \{x \in \mathcal{X}_i^{\mathcal{L}} | L(x) = l_j\}$). □

Figure 2(e) shows four movie objects (M). If M1 and M3 are labeled "♦", M2 is labeled "★", and M4 is unlabeled, then there are two label-induced clusters (indicated by dotted ovals in Figure 2(g)). Note that since M4 is unlabeled, it does not belong to any cluster.

3. RELATED WORK

Classification on networked data has been well studied in the past decade [7, 10, 5, 25]. A number of transductive classifiers have been proposed, especially on homogeneous information networks [9, 3, 24, 1, 2, 19, 26] and specific HINs [20]. Other non-transductive classifiers on general HINs include HINAL [17], HCC [6], etc.

There are also a few transductive classifiers for classifying objects in general HINs. Three representatives of such algorithms are GNetMine [4], HetPathMine [8] and Grempt [18]. As we have explained in Section 1, our goal is not to evaluate the performance of individual transductive classifiers. Rather, these classifiers serve as tools for us to analyze the intrinsic properties of HINs in the context of transductive classification. Our coverage of transductive classifiers is therefore not meant to be exhaustive.

GNetMine is among the first methods proposed for HIN object classification. GNetMine first constructs a predictive function $f(l_j|x)$ for each object x and object label l_j. It then derives an objective function that captures the assumption that highly related objects should be given similar labels. This is achieved by minimizing two values: (1) for any two highly related objects x_p and x_q (as reflected by their edge connections in the HIN), the difference between their predictive values $f(l_j|x_p)$ and $f(l_j|x_q)$, and (2) for any labeled object, x_r, the difference between its predictive value $f(l_j|x_r)$ and its true label-induced value, which is 1 if x_r's label is l_j; 0 otherwise. The predictive functions $f(l_j|x)$'s are trained by optimizing the objective function via an iterative method. Finally, GNetMine makes label predictions based on the $f(l_j|x)$'s.

Meta-path has been successfully used in various data mining tasks. These include cluster analysis [14], recommender systems [22], link prediction [23], and object similarity search [21]. HetPathMine takes advantage of meta-path to perform transductive classification. The basic idea is to first transform an HIN into a number of TSSNs (e.g., in Figure 1, the HIN is converted into three TSSNs). It also derives an objective function to minimize the two values mentioned in GNetMine. However, the function aims to minimize the difference between predictive values of any two highly related objects in each TSSN instead of the original HIN. Weights are assigned to the TSSNs and the function makes a weighted combination on these homogeneous networks. These weights are learned by optimizing a cost function.

Grempt is a graph regularized transductive regression model. The objective is to predict numerical values of objects in an HIN with transductive learning. It shares similar ideas with HetPathMine in (1) using meta-paths to derive TSSNs, (2) formularizing objective function based on the TSSNs, and (3) learning the weights of TSSNs in order to make a weighted combination on them. Grempt can be easily adapted to perform classification by considering the task as a regression on the predictive values $f(l_j|x)$'s. An interesting aspect of Grempt is that it formulates the learning problem as a convex optimization problem by applying a constraint function with which optimal weights are learned. This feature makes Grempt an effective method.

4. ANALYSIS

In this section we analyze the structural properties of HINs and classification tasks. We first describe three classification tasks, which are used in our analysis. Then, we explain the concepts of cohesiveness and connectedness, and propose quantitative measures to capture these properties.

4.1 Classification tasks

DBLP[3] is a bibliographic information network. We extracted a dataset from DBLP that contains 14,376 papers (P), 20 publication venues (V), 14,475 authors (A) and 8,920 terms (T). These form the objects of the HIN. There are three types of links, which are authorship (A-P), publication (P-V), and keyword (P-T). The task is to classify authors into their research areas. The label set is {*database* (DB), *data mining* (DM), *artificial intelligence* (AI) and *information retrieval* (IR) }. We use the set of meta-paths {APA, APAPA, APVPA, APTPA} as suggested in [14].

Yago Movie is a movie related HIN extracted from Yago. The dataset contains 1,465 movies (M), 4,019 actors (A), 1,093 directors (D) and 1,458 writers (W). There are three types of links: M-A, M-D, and M-W. All of the extracted movies can be classified into one of three genres: *horror*, *action* and *adventure*. The task is to label movies into their genres. We use the meta-path set {MAM, MDM, MWM, MAMAM, MDMDM, MWMWM} as suggested in [17].

Freebase Movie is another movie related HIN extracted from Freebase. It consists of 3,492 movies (M), 33,401 actors (A), 2,502 directors (D) and 4,459 producers (P). There are three types of links: M-A, M-D, and M-P. The task is again to label movies into their genres. The label set is {*action*, *adventure* and *crime*}. We use the meta-path set {MAM, MDM, MPM, MAMAM, MDMDM, MPMPM} [17].

Table 3: Similarity (NMI) of $\mathcal{C}_{\hat{L}}$ and $\mathcal{C}_{NetClus}$

DBLP	Yago Movie	Freebase Movie
0.707	0.018	0.027

4.2 Structural properties of an HIN

Transductive classification on networked data utilizes links and paths to evaluate the relatedness of objects. Objects that are highly related are assumed to share similar labels. In a sense, links and paths are used to propagate labels from labeled objects to unlabeled ones. Algorithms like GNetMine use the HIN to propagate labels, while algorithms like HetPathMine and Grempt use meta-paths to derive TSSNs and propagate labels on those sub-networks. In any case, the network structure (of the original HIN or of the derived TSSNs) is an important factor of the effectiveness of transductive classifiers. In particular, all these transductive classifiers share the following intrinsic assumption:

Assumption 1. **The Connectivity Assumption**. *The structural connectivity between two objects (via links and paths) is highly correlated to whether the objects would share the same label.*

In this section we address two interesting questions:

Question 1: Does the connectivity assumption generally hold for HIN classification tasks?

Question 2: If not, how to measure the validity of the connectivity assumption given a classification task?

To answer the first question, we conducted simple experiments on a number of classification tasks. First, we define *true-labeling* using the notations of Definition 5:

Definition 7. **True-labeling**. A labeling $\hat{L}$ is a true-labeling if $\forall x \in \mathcal{X}_i$, $\hat{L}(x)$ is the true label (ground truth) of object x.

We put a caret '∧' on a labeling L to indicate that it is a true-labeling. For each of the three classification tasks DBLP, Yago Movie and Freebase Movie, we find the true-labeling $\hat{L}$. We then cluster objects into the label-induced clustering $\mathcal{C}_{\hat{L}}$ (see Definition 6). Each cluster in $\mathcal{C}_{\hat{L}}$ thus contains all and only those objects of a given label.

Next, we apply NetClus [12], which is a clustering method that clusters objects in an HIN based on network structure, to our HINs. For each HIN, we compare the true-label-induced clustering $\mathcal{C}_{\hat{L}}$ (which is based solely on object labels) with the clustering $\mathcal{C}_{NetClus}$, given by NetClus (which is based solely on network structure). The similarity of the two clusterings is measured by normalized mutual information (NMI). Table 3 shows the results. We see that for DBLP, the NMI is high, indicating that $\mathcal{C}_{\hat{L}}$ and $\mathcal{C}_{NetClus}$ are highly similar. In other words, objects that are highly connected (put in the same cluster by NetClus) tend to share the same label (put in the same cluster by the true-label-induced clustering). The connectivity assumption is thus strongly valid. On the other hand, for Yago Movie and Freebase Movie, the NMI's are very low, indicating that the connectivity assumption does not hold in those cases. This analysis is consistent with the accuracies of the transductive classifiers when they are applied to the three classification tasks (see Table 1). Our analysis leads to the following conclusion:

[3] http://dblp.uni-trier.de/

Conclusion 1. *The connectivity assumption does not always hold across HIN classification tasks. When it does, transductive classifiers are very effective.*

The next question we address is how the validity of the connectivity assumption is evaluated. We propose to measure the correlation between *structural connectivity of objects* and their *label similarity* by the concepts of cohesiveness and connectedness. Intuitively, given a classification task, an HIN is *highly cohesive* if strong connectivity occurs mostly between objects of the same label; and that the HIN is *highly connected* if objects of the same label exhibit strong connectivity. The correlation between structural connectivity and label similarity is thus high if the HIN is highly cohesive and highly connected. In the following discussion, we first assume that the true-labeling $\hat{L}$ of a classification task is known. Cohesiveness and connectedness are then defined based on a true-labeling. We will discuss in Section 5.1 how the two measures can be estimated when the true-labeling is not known in practice.

4.3 Cohesiveness

Given an HIN $G = (V, E)$, consider the task of classifying objects $x \in \mathcal{X}_i$ of type T_i with the label set $\mathcal{L} = \{l_1, ..., l_k\}$. A transductive classifier propagates label from an object $x_u \in \mathcal{X}_i$ to another object $x_v \in \mathcal{X}_i$. How much this propagation is done depends on the structural connectivity (i.e., links and paths) between x_u and x_v. For example, HetPathMine and Grempt use meta-paths to derive TSSNs (see Figure 1), and for each TSSN $G_{\mathcal{P}}$, which is derived from a meta-path $\mathcal{P}$, the structural connectivity between x_u and x_v is measured using *PathSim* [13]:

$$s(x_u, x_v) = \frac{2 \times |\{p_{x_u \rightsquigarrow x_v} : p_{x_u \rightsquigarrow x_v} \vdash \mathcal{P}\}|}{|\{p_{x_u \rightsquigarrow x_u} : p_{x_u \rightsquigarrow x_u} \vdash \mathcal{P}\}| + |\{p_{x_v \rightsquigarrow x_v} : p_{x_v \rightsquigarrow x_v} \vdash \mathcal{P}\}|}.$$

Now, let us consider the true-label-induced clustering $\mathcal{C}_{\hat{L}}$. Figure 3 shows an example clustering with 2 clusters (objects with true label '○' and those with true label '□'). An edge, e.g., (x_1,x_2), is shown to indicate that two objects are structurally connected (e.g., they are connected by meta-paths). We assume that each edge shown is associated with a weight, which reflects the strength of the connection (e.g., as measured using *PathSim*). We call Figure 3 a *structural connectivity graph.*

Note that transductive classifiers using meta-paths (such as HetPathMine and Grempt) measure the connectivity between two objects based on how well the objects are connected by meta-paths. In this case, the structural connectivity graph can be seen as a composition (union) of the TSSNs derived from a given set of meta-paths. For example, the structural connectivity graph of the movie HIN shown in Figure 1(a) is the composition of the TSSNs shown in Figures 1(c)-(e). In particular, the edge connecting M1 and M2 in Figure 1(c) indicates that M1 and M2 are structurally connected by a meta-path instance (M1-A1-M2). If M1 is labeled, the label will be propagated to M2 via the structural connection (M1,M2). We will discuss how an overall cohesiveness value is measured when the structural connectivity graph is a composition of multiple TSSNs shortly. For the moment, let us assume that there is only one connectivity graph derived.

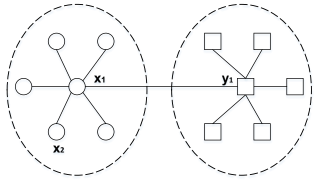


Figure 3: A structural connectivity graph

As mentioned, an HIN is highly cohesive if strong connectivity occurs mostly between objects of the same label. Referring to Figure 3, that means intra-cluster edges are many-and-strong, while inter-cluster edges are few-and-weak. Figure 3 shows a very cohesive HIN because there is only one edge (x_1, y_1) across the two clusters; most of the structural connections are between objects of the same label. With this intuition, we quantitatively define cohesiveness as follows.

Given two clusters C_1 and C_2 in a true-label-induced clustering $\mathcal{C}_{\hat{L}}$, with respect to a structural connectivity graph, let h_1 (h_2) be the number of intra-cluster edges in C_1 (C_2) with a sum of edge weights w_1 (w_2). Also, let $h_{1,2}$ be the number of inter-cluster edges between C_1 and C_2 with a sum of edge weights $w_{1,2}$. Define,

$$\rho(C_1) = \frac{h_1}{h_{1,2} + h_1}, \quad \rho(C_2) = \frac{h_2}{h_{1,2} + h_2}, \text{ and} \tag{1}$$

$$\eta(C_1) = \frac{w_1}{w_{1,2} + w_1}, \quad \eta(C_2) = \frac{w_2}{w_{1,2} + w_2}. \tag{2}$$

$\rho(C_1)$ can be interpreted as, "Among all the edges that connect some objects in C_1, the fraction of which that connect *only* objects in C_1." The other quantities can be interpreted similarly. We further define the *pairwise cluster cohesiveness* of C_1 and C_2:

$$\Upsilon(C_1, C_2) = \rho(C_1) \times \rho(C_2) \times \eta(C_1) \times \eta(C_2). \tag{3}$$

For a classification task with k labels, a labeling induces k clusters $C_1, ..., C_k$. Let $b_i = |C_i|$. Define the *cluster cohesiveness* of C_i by

$$\Upsilon_{C_i} = \frac{1}{k-1} \sum_{j \neq i} \Upsilon(C_i, C_j). \tag{4}$$

Let $\boldsymbol{\Upsilon} = (\Upsilon_{C_1}, ..., \Upsilon_{C_k})^{\mathrm{T}}$. We define the cohesiveness of an HIN G as the weighted average of the cluster cohesiveness:

$$\Upsilon_G = \boldsymbol{\beta}\boldsymbol{\Upsilon}, \tag{5}$$

where $\boldsymbol{\beta} = (\frac{b_1}{\sum_{i=1}^k b_i}, \frac{b_2}{\sum_{i=1}^k b_i}, ..., \frac{b_k}{\sum_{i=1}^k b_i})$.

If we use a set of meta-paths $\mathcal{P}_1, ..., \mathcal{P}_r$ in transductive classification, the structural connectivity graph can be seen as a composition of a number of TSSNs $G_{\mathcal{P}_j}$ $(1 \leq j \leq r)$. In this case, we evaluate the cohesiveness of each TSSN to obtain $\Upsilon_{G_{\mathcal{P}_j}}$, assign a weight θ_j to each meta-path $\mathcal{P}_j$, and the overall cohesiveness is given by the weighted average:

$$\Upsilon_G = \sum_{j=1}^{r} \theta_j \Upsilon_{G_{\mathcal{P}_j}}. \tag{6}$$

We assume that the weights θ_j's can be learned. Due to space limitation, readers can refer to [8, 18, 17] for some example methods for learning meta-path weights. In this paper, we will rely on Grempt to learn the weights.

Table 4: Cohesiveness of HIN classification tasks

DBLP: $\Upsilon_{DBLP} = 0.536$						
$\mathcal{P}$	APA	APAPA	APVPA	APTPA		
$\Upsilon_{G_{\mathcal{P}}}$	0.733	0.483	0.393	0.016		

Yago: $\Upsilon_{Yago} = 0.209$						
$\mathcal{P}$	MAM	MDM	MWM	MAMAM	MDMDM	MWMWM
$\Upsilon_{G_{\mathcal{P}}}$	0.106	0.313	0.262	0.065	0.303	0.214

Freebase: $\Upsilon_{Freebase} = 0.185$						
$\mathcal{P}$	MAM	MDM	MPM	MAMAM	MDMDM	MPMPM
$\Upsilon_{G_{\mathcal{P}}}$	0.107	0.326	0.174	0.086	0.346	0.123

We computed the cohesiveness values of the three HIN classification tasks. Table 4 shows the results. We see that DBLP has a much higher cohesiveness value (Υ_{DBLP} = 0.536) compared with Yago (Υ_{Yago} = 0.209) and Freebase ($\Upsilon_{Freebase}$ = 0.185). Again, this is consistent with our analysis that the connectivity assumption is more valid with DBLP than with Yago or Freebase. In Table 4, we also show the cohesiveness values of the TSSNs derived from various meta-paths. For example, for DBLP, the TSSN G_{APA}, derived from the meta-path APA, is much more cohesive than those given by other meta-paths. The interpretation is that co-authorship (which is captured by the meta-path APA) occurs mostly between authors of the same area. On the other hand, the small cohesiveness value of G_{APTPA} indicates that authors of different areas could share the same keywords in their papers. For Yago Movie and Freebase Movie, the meta-paths MDM and MDMDM derive the most cohesive TSSNs. Yet, their cohesiveness values are much smaller than that of APA, suggesting that it is more difficult for transductive classification to achieve high accuracy in classifying movies.

4.4 Connectedness

We say that an HIN is highly connected if objects of the same label exhibit *strong connectivity*. With respect to transductive classification, this connectivity should facilitate label propagation from one object to another of the same class. To illustrate the idea, consider Figure 4, which shows two object clusters (○ and □) in a structural connectivity graph. We see that objects in the □ cluster are strongly connected in the sense that if an object in the cluster (say y_1) is labeled, the label can be propagated effectively to all other objects in the same cluster. The ○ cluster, on the other hand, is less connected. In particular, if we consider only the intra-cluster edges of the ○ cluster, the ○ objects form two isolated components. If object x_1 in component 1 is labeled, the label cannot be propagated to the objects in component 2 (e.g., x_2) without going through the □ cluster. Label propagation among the ○ objects is thus less effective.

We measure the connectedness of a cluster C by the number of disconnected components ($NDC(C)$) in C if only intra-cluster edges are considered. For example, in Figure 4, the NDC of the ○ cluster is 2, while that of the □ cluster is 1. The larger $NDC(C)$ is, the less is the connectedness of cluster C. We normalize this measure to [0,1] and define *cluster connectedness*, Ψ_C:

$$\Psi_C = \begin{cases} 1 & \text{when } NDC(C) = 1, \\ 1 - \frac{NDC(C)}{b} & \text{when } NDC(C) > 1, \end{cases} \tag{7}$$

where b is the number of objects in C.

If there are k clusters $C_1, ..., C_k$, corresponding to k labels of a classification task, let $\boldsymbol{\Psi} = (\Psi_{C_1}, \Psi_{C_2}, ..., \Psi_{C_k})^{\mathrm{T}}$. We define the connectedness of an HIN G as the weighted average of the cluster connectedness:

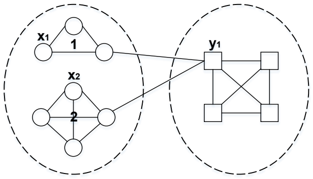


Figure 4: An example illustrating connectedness

Table 5: Connectedness of HIN classification tasks

DBLP: $\Psi_{DBLP} = 0.942$						
$\mathcal{P}$	APA	APAPA	APVPA	APTPA		
$\Psi_{G_{\mathcal{P}}}$	0.899	0.920	1.0	1.0		

Yago: $\Psi_{Yago} = 0.393$						
$\mathcal{P}$	MAM	MDM	MWM	MAMAM	MDMDM	MWMWM
$\Psi_{G_{\mathcal{P}}}$	0.567	0.253	0.281	0.690	0.253	0.285

Freebase: $\Psi_{Freebase} = 0.584$						
$\mathcal{P}$	MAM	MDM	MPM	MAMAM	MDMDM	MPMPM
$\Psi_{G_{\mathcal{P}}}$	0.970	0.282	0.350	0.992	0.282	0.382

$$\Psi_G = \boldsymbol{\beta}\boldsymbol{\Psi}, \tag{8}$$

where $\boldsymbol{\beta} = (\frac{b_1}{\sum_{i=1}^{k} b_i}, \frac{b_2}{\sum_{i=1}^{k} b_i}, ..., \frac{b_k}{\sum_{i=1}^{k} b_i})$.

Similar to our discussion of cohesiveness, if we use meta-paths $\mathcal{P}_1, ..., \mathcal{P}_r$ in transductive classification, we evaluate $\Psi_{G_{\mathcal{P}_j}}$ for each TSSN $G_{\mathcal{P}_j}$. The overall connectedness of an HIN G is the weighted average:

$$\Psi_G = \sum_{j=1}^{r} \theta_j \Psi_{G_{\mathcal{P}_j}}, \tag{9}$$

where θ_j's are the meta-path weights.

Table 5 shows the connectedness values of our classification tasks. The connectedness of the TSSN derived from each meta-path considered is also shown. From the table, we see that DBLP has a much higher connectedness value (0.942) compared with Yago (0.393) and Freebase (0.584). This means that authors of the same area mostly form a single structurally connected component. The label of one author can therefore be very effectively propagated to other authors of the same area via meta-paths. Comparing the four meta-paths used in DBLP, we see that the connectedness values of APVPA and APTPA are even higher than that of APA. The interpretation is that authors of the same area tend to attend the same conferences and use similar keywords in their papers, but they do not necessarily co-author with each other. For Yago Movie and Freebase Movie, we see that MAM and MAMAM give relatively high connectedness values, indicating that movies of the same genre tend to be starred by the same actors. However, the two movie HINs are generally much less connected than DBLP.

Now, let us study how cohesiveness and connectedness are correlated to classification accuracy. We apply Grempt on the three classification tasks. For each one, we further obtain the accuracy when only one meta-path (and its derived TSSN) is used. Table 6 shows the results. For example, if Grempt uses only the meta-path APA to derive the structural connectivity between objects in DBLP, the classification accuracy is 42.8%; If all four meta-paths are considered, then Grempt achieves an accuracy of 89.3%. Note that for DBLP, the training set is much smaller (0.5%) than that of Yago Movie and Freebase Movie (5%).

From Tables 4, 5, 6, we draw the following observations:

Table 6: Accuracies of applying Grempt to HINs

DBLP: 0.5% labeled objects, classification accuracy = 89.3%						
$\mathcal{P}$	APA	APAPA	APVPA	APTPA		
acc.	42.8%	44.0%	91.1%	35.3%		

Yago: 5% labeled objects, classification accuracy = 49.2%						
$\mathcal{P}$	MAM	MDM	MWM	MAMAM	MDMDM	MWMWM
acc.	41.5%	4.8%	8.8%	41.3%	4.8%	8.7%

Freebase: 5% labeled objects, classification accuracy = 65.4%						
$\mathcal{P}$	MAM	MDM	MPM	MAMAM	MDMDM	MPMPM
acc.	65.7%	6.1%	14.1%	66.0%	6.1%	14.4%

(1) The cohesiveness and connectedness of DBLP are both much higher than those of Yago and Freebase, and the classification accuracy of DBLP (89.3%) is also much higher than those of Yago (49.2%) and Freebase (65.4%).
(2) For DBLP, the meta-path APVPA gives the highest accuracy (91.1%). This is because its TSSN is the most connected (1.0) and is reasonably cohesive (0.393, which is higher than any cohesiveness values in Yago or Freebase).
(3) The accuracy of the TSSN due to meta-path APTPA (35.3%) is much lower than that of APVPA (91.1%) although both of them are perfect in their connectedness scores (1.0). The reason is that the cohesiveness value of APTPA is extremely low (0.016). This indicates that, with APTPA, although a label propagates well among objects within the same cluster (high connectedness), the label also propagates over to other clusters as well (very low cohesiveness).
(4) For Yago and Freebase, although MDM and MDMDM give relatively cohesive TSSNs (Υ: 0.303-0.346) among all meta-paths for the two tasks, the TSSNs are highly disconnected (Ψ: 0.253-0.282). This explains why the classification accuracies using only MDM or MDMDM are so poor (6.1%).

From these observations, we can conclude that cohesiveness and connectedness are highly correlated with classification accuracy. Also, both factors are important to the successful application of transductive classifiers.

5. ESTIMATING Υ AND Ψ

We measure cohesiveness (Υ_G) and connectedness (Ψ_G) assuming that a true-labeling $\hat{L}$ on the set of objects $\mathcal{X}_i$ to be classified is available. The labeling induces a true-label-induced clustering $C_{\hat{L}}$ based on which structural connections are categorized into intra- and inter-cluster edges in the structural connectivity graph. Cohesiveness and connectedness are then defined based on such edges. In practice, however, such a true labeling is unavailable. In this section we discuss how the two measures can be practically estimated. We also discuss how the estimated values allow us to design (1) a black-box tester, which indicates whether transductive classification is generally successful for the classification task, and (2) an effective active learning algorithm.

We use a simple "bootstrapping" approach to estimate Υ_G and Ψ_G: Given a set of training data D (i.e., a set of objects whose true labels are known), we first apply a classifier A (e.g., Grempt) on G to obtain a labeling L_A. L_A partitions the object set $\mathcal{X}_i$ into two sets: the set of labeled objects $\mathcal{X}_i^{\mathcal{L}}$ and the set of unlabeled objects $\mathcal{X}_i^*$ (see Definiton 5). We first ignore the objects in $\mathcal{X}_i^*$. Then, we obtain the label-induced clustering, $\mathcal{C}_{L_A}$, on the objects in $\mathcal{X}_i^{\mathcal{L}}$. We use $\mathcal{C}_{L_A}$ as an approximation of the true-label-induced clustering $C_{\hat{L}}$ and compute an approximated cohesiveness (denoted by Υ'_G) and an approximated connectedness (denoted by Ψ'_G). Since the label of any object $x \in \mathcal{X}_i^*$ cannot be deduced by the classifier A, x must not be connected to any labeled objects in D via any edges or paths (and hence labels cannot be propagated to x). A larger set $\mathcal{X}_i^*$ thus indicates weaker structural connectivity. Hence, we assess a penalty on Υ'_G and Ψ'_G by a discount factor $df = 1 - |\mathcal{X}_i^*|/|\mathcal{X}_i|$. That is, we assign $\Upsilon'_G \leftarrow df \cdot \Upsilon'_G$ and $\Psi'_G \leftarrow df \cdot \Psi'_G$.

5.1 Black-box tester

Given an HIN classification task, our black-box tester first computes the estimated Υ'_G and Ψ'_G. These values are then compared with certain standard references, which are HIN classification tasks with known Υ_G, Ψ_G and accuracy. The tasks we studied in this paper, namely, DBLP, Yago Movie, and Freebase Movie are examples of such references. Specifically, if the estimated (Υ'_G, Ψ'_G) of a given task are both strictly better than those of, say, DBLP (0.536, 0.942), we have good confidence that transductive classifiers will be effective because DBLP is shown to be a "good" case of transductive classification. Conversely, if the estimated values are worse than those of say Yago Movie, transductive classification is unlikely to perform well for the task. The advantage of the black-box tester over a direct evaluation of the accuracies of transductive classifiers is that the black-box tester does not require a test set. This is particularly useful for HINs where labeled data is generally hard to come by.

Algorithm 1 ALCC

```
Input: $G$, $A$, $\mathcal{X}_i$, $\mathcal{L}$, $D$, $B$, $N_s$.
Output: $\Delta_D$
$\Delta_D \leftarrow \emptyset$
Execute $A$ on $G$ with $D$ to obtain a labeling $L_A$
Compute $\Upsilon'_G$, $\Psi'_G$ based on $L_A$
for $k = 1$ to $B/N_s$ do
    for $(x \in \mathcal{X}_i - D)$, $(l_j \in \mathcal{L})$ do
        Get $D_{+(x,j)}$
        Execute $A$ on $G$ with $D_{+(x,j)}$; Compute $\Upsilon'_G$, $\Psi'_G$
        Compute $QS(D_{+(x,j)})$
    end for
    Compute $QS(D_{+(x,*)})$
    $S_k \leftarrow N_s$ objects with largest $QS(D_{+(x,*)})$
    $D \leftarrow D \cup S_k$; $\Delta_D \leftarrow \Delta_D \cup S_k$
end for
return $\Delta_D$
```

5.2 Active learning

Our method of estimating Υ'_G and Ψ'_G also leads to an interesting approach to active learning in HIN classification. Given a budget B, the problem of active learning is to select a set Δ_D of B objects in $\mathcal{X}_i - D$ to obtain their labels. These objects (with their acquired labels) are then added to the training set D. Active learning is about finding the best set Δ_D so as to achieve the largest improvement in classification accuracy with the expanded training set $D \cup \Delta_D$.

Our active learning algorithm ALCC aims at finding the B objects that give the largest improvement in Υ'_G and Ψ'_G. Specifically, we define $QS(D) = \Upsilon'_G \times \Psi'_G$ as the *quality score*, where Υ'_G and Ψ'_G are computed w.r.t. a training set D. When we estimate Υ'_G and Ψ'_G, a transductive classifier A is applied. Generally, for each object $x \in \mathcal{X}_i^{\mathcal{L}}$, A determines a *label distribution* $(f_x^1, f_x^2, ..., f_x^k)$ for x, where each component f_x^j represents the confidence that x should be assigned the label l_j. Also, for each object $x \in \mathcal{X}_i^*$, f_x^j is given by the prior probability $f_x^j = |\{y \in D | \hat{L}(y) = l_j\}|/|D|$. We use $D_{+(x,j)}$ to denote the expanded training set if object x were given the label l_j and is added to D. We measure the expected quality score if object x is added to the training set by:

$$QS(D_{+(x,*)}) = \sum_{j=1}^{k} f_x^j \times QS(D_{+(x,j)}). \qquad (10)$$

ALCC picks N_s objects x's that give the best expected quality scores $QS(D_{+(x,*)})$ and adds these objects to D. This process is repeated B/N_s times until the budget is exhausted. Algorithm 1 shows the pseudo code of ALCC.

6. EXPERIMENTS

We conducted experiments to analyze our method of estimating Υ'_G and Ψ'_G. We discuss the effectiveness of the black-box tester and present case studies of applying the tester to HIN classification tasks. Finally, we evaluate the effectiveness of our active learner ALCC and compare its performance against other active learning methods.

6.1 Results

We apply our estimation method to DBLP, Yago Movie and Freebase Movie. Figures 5(a) and (b) show the estimated Υ'_G and Ψ'_G, respectively, as the training set D varies from 5% to 100% of the object set $\mathcal{X}_i$. Note that when $|D|$ = 100%, all the object labels are known and hence the estimated values are the true values of Υ_G and Ψ_G (which are also shown in Tables 4 and 5). For reference, Figure 5(c) shows the classification accuracy of Grempt when it is applied to the tasks under various sizes of the training set.

From Figure 5(c), we see that for DBLP, the classification accuracy is very high. As a result, the labeling L_A we used in estimating Υ'_G and Ψ'_G is very close to the true labeling, which in turn, induces a clustering that is highly similar to the true-label-induced clustering $\mathcal{C}_{\hat{L}}$. This results in very accurate estimation of Υ'_G and Ψ'_G as reflected by the fairly flat curves for DBLP in Figures 5(a) and (b); the estimated values are close to the rightmost points (at $|D|$ = 100%, the case of true values).

The classification accuracies for Yago and Freebase are quite low. For example, when $|D|$ = 5%, their accuracies are 49% and 65%, respectively. From Figure 5(b), we see that the curves of the estimated Ψ'_G for Yago and Freebase are both very flat. This indicates that despite low classification accuracy, our estimation of connectedness remains highly accurate. Cohesiveness, on the other hand, is harder to accurately estimate when the classification accuracy is low. For example, when $|D|$ = 5%, the estimated Υ'_{Yago} is 0.317 while the true value is 0.209.

We have conducted numerous experiments on HIN classification tasks. Generally, for those tasks on which transductive classifiers perform poorly (such as Yago Movie and Freebase Movie), our estimated values Υ'_G are over-estimates. This over-estimation lessens as we get more training data. For example, in Figure 5(a), we see that the curves for Yago and Freebase generally go down as $|D|$ increases towards 100%; All the estimated values are above the true values.

The phenomenon of overestimating Υ'_G can be explained by the illustration shown in Figure 6. A reason why transductive classification performs poorly on an HIN is that the HIN exhibits poor cohesiveness. Figure 6(a) shows the structural connectivity graph of a non-cohesive HIN — there are quite a few (4) inter-cluster connections relating objects of different clusters (labels). Given a small training set (illustrated as filled objects in Figure 6(b)), our estimation method applies a transductive classifier to propagate labels. Because of the high number of inter-cluster connections, labels from one cluster can be easily propagated to objects of another cluster, resulting in mis-labeling. (This is illustrated in Figure 6(c) where object y_4 is mislabeled ●.) The resulting inferred clustering structure is shown in Figure 6(d). Since the HIN is not cohesive, intra-cluster connections could be weak. The fact that object y_4 is mislabeled by the classifier indicates it is not strongly connected to other objects of the □ label. The number of inter-cluster connections found in the inferred clustering (2) (Figure 6(d)) is thus smaller than that of the true clustering (4) (Figure 6(a)). With fewer inter-cluster connections, the cohesiveness given by the inferred clustering is thus overestimated.

Recall that our black-box tester first estimates Υ'_G and Ψ'_G for a given HIN G, and then compares the estimates against the (Υ_G, Ψ_G) values of standard references. The tester can make one of two recommendations: Case 1: (Υ'_G, Ψ'_G) are strictly better than the values of a "good case" such as DBLP. In this case, the tester recommends transductive classification be applied to the task. Since the tester estimates that G is highly cohesive and highly connected (even better than a "good case" of standard references), from our previous discussion, we know that the estimated (Υ'_G, Ψ'_G) should be highly accurate. The recommendation is therefore reliable. Case 2: (Υ'_G, Ψ'_G) are strictly worse than the values of a "bad case" such as Yago Movie. In this case, the tester does not recommend transductive classification be applied. Since the estimated values are worse than a "bad case", from our discussion, the estimated (Υ'_G, Ψ'_G) are likely to be overestimates. That is, the true values should be even worse. The recommendation of the tester (of not using transductive classification) is therefore reliable as well.[4]

6.2 Case study

We applied our estimator to a number of other HIN classification tasks. In this section we further present two representative cases. We will also discuss applying the black-box tester to these cases to make a recommendation. First, let us describe the two classification tasks.

[TV] We extracted an HIN from Freebase on objects that are related to TV program series. The HIN consists of 2,913 series (S), 652 directors (D), 685 writer (W), and 151 TV programs (P). The schematic graph consists of three types of links, namely, series-director, series-program, and series-writer. All the series objects extracted are of one of the three genres: *comedy-drama*, *soap opera*, and *police procedural*. The classification task is to classify series objects into their genres. We consider 6 meta-paths: {SDS, SWS, SPS, SDSDS, SWSWS, SPSPS}.

[Game] The other HIN we extracted from Freebase is related to video games. The HIN consists of 4,095 games (G), 1,578 publishers (P), 2,043 developers (D) and 197 designers (S). The schematic graph consists of three types of links: game-publisher, game-developer, and game-designer. All the game objects extracted are of one of the three genres: *action*, *adventure*, and *strategy*. The classification task is to classify games into their genres. We consider 6 meta-paths: {GPG, GDG, GSG, GPGPG, GDGDG, GSGSG}.

[4] For cases in which (Υ'_G, Ψ'_G) are neither strictly better than some good cases nor strictly worse than some bad cases, the tester does not make a recommendation.

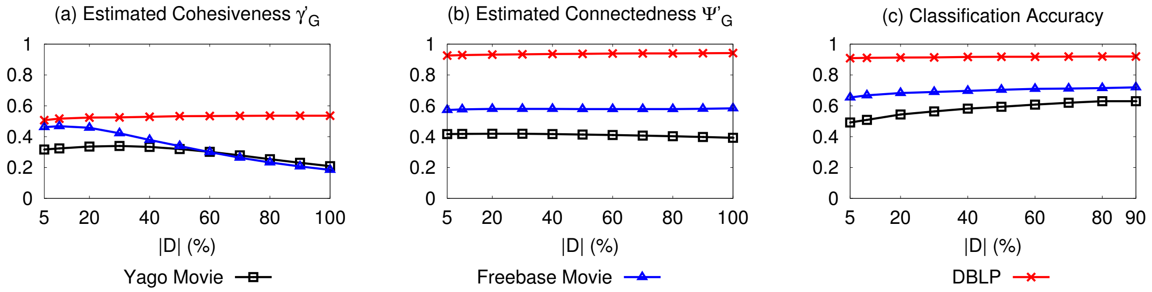
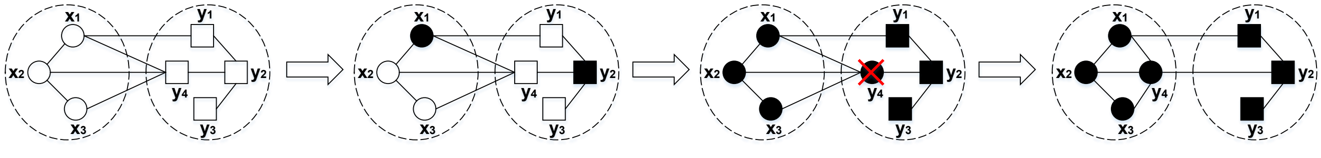


Figure 5: Estimating cohesiveness, connectedness, and classification accuracy of 3 HIN classification tasks



(a) true clustering structure (b) training set D = {**x1**, **y2**} (c) classification result: **y4** misclassified (d) inferred clustering structure

Figure 6: Overestimation of Υ'_G for a non-cohesive HIN

We first apply our estimator to TV with a training set D of 15% of all the program series objects (S). The estimator returns ($\Upsilon'_{TV} = 0.749$, $\Psi'_{TV} = 0.836$). These values show that TV is highly cohesive and highly connected for the classification task. Assuming that we have only DBLP, Yago Movie, and Freebase Movie as standard references. We compare (Υ'_{TV}, Ψ'_{TV}) against those of the references. We found that (1) TV is not strictly worse than the *bad cases*: Yago Movie (0.209, 0.393) or Freebase Movie (0.185, 0.584), and (2) TV is comparable to but not strictly better than the *good case*: DBLP (0.536, 0.942). In fact, the estimator indicates that TV is more cohesive than DBLP but not as connected. For TV, although the black-box tester does not give a recommendation, we have high confidence that transductive classification will be successful because of the high estimated values. We then obtain the true labels of all the series objects S. We apply Grempt to classify the series objects (with a 15% training set) and compare the labels predicted by Grempt against the true labels. We found that the classification accuracy is 94.3%, showing that transductive classification is indeed highly accurate for TV. Furthermore, with the true labels, we compute the true cohesiveness and connectedness of TV, which are (0.887, 0.889). Our estimated values, (0.749, 0.836), are quite close to the true values. With this analysis, we add [TV: (0.887, 0.889)] as a good case to our set of standard references.

Next, we apply our estimator to Game with a training set D of 15% of all game objects (G). The estimator returns ($\Upsilon'_{Game} = 0.342$, $\Psi'_{Game} = 0.254$). Although the estimated values are not strictly worse than those of Yago Movie or Freebase Movie, the small values indicate that Game is likely a "bad" case of transductive classification. Again, we obtain the true labels of all game objects and found that the accuracy of applying Grempt to Game is only 34.2%. We further determine the true cohesiveness and connectedness values of Game, which are ($\Upsilon_{Game} = 0.250$, $\Psi_{Game} = 0.297$). Note that $\Upsilon'_{Game} > \Upsilon_{Game}$. This is consistent with our discussion that for "bad" cases, the estimated cohesiveness values are generally overestimates. We add [Game: (0.250, 0.297)] as a bad case to our set of standard references. As more HIN tasks are added to the references, the black-box tester accumulates more examples to refine its recommendations.

6.3 Active learning

In this section we evaluate our active learning algorithm ALCC. We compare ALCC against three other methods:
Random: Given a budget B and an initial training set D, *Random* randomly picks B objects in $X_i - D$.
Uncertainty Sampling (US): Recall that for each object $x \in X_i - D$, a classification algorithm assigns to it a label distribution $(f_x^1, f_x^2, ..., f_x^k)$, where f_x^j is the likelihood that object x is of label l_j (see Section 5.2). US evaluates the entropy of each object's label distribution and picks B objects whose entropies are the largest.
Active Learning based on Global Entropy (ALGE): Recall that ALCC computes a quality score $QS(D) = \Upsilon'_G \times \Psi'_G$ of an HIN G given a training set D, and essentially picks B objects that can best improve the score. In order to evaluate the importance of cohesiveness and connectedness in active learning, we consider an alternative definition of QS. Specially, we define $QS(D)$ = the average entropy of the label distributions of all the objects $x \in X_i - D$. We modify ALCC with the above definition of QS and call the resulting algorithm ALGE.

We execute the four active learning algorithms on a number of HIN classification tasks. As representative results, Figure 7 shows the algorithms' performance when they are applied to Yago Movie (Figure 7(a)) and Game (Figure 7(b)). For both HINs, the initial training set D is set to 10% of the object set. We vary the budget B from 0 to 10%. For ALCC and ALGE, N_s is set to 2. We use Grempt as the classification algorithm and report its accuracy after D is expanded with the objects picked by the active learning algorithms.

From Figure 7, we see that as the budget B increases, we get more labeled objects in the training set and classification accuracy increases. Comparing the four algorithms, Random gives the worst performance. US, which is based on local per-object entropy, generally performs worse than ALGE, which considers the overall entropy among all the objects. This is particularly true for the Game HIN. ALCC,

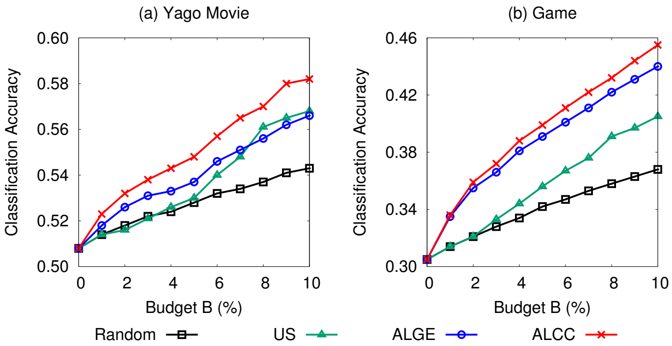


Figure 7: Active learner comparison

which employs cohesiveness and connectedness measures, gives the best performance among the four. For example, when $B = 10\%$, for Yago Movie, ALCC improves the accuracy by +7.4% (from 50.8% to 58.2%). This compares favorably against Random (+3.5%), US (+6%), and ALGE (+5.8%). For Game, although ALCC only registers a marginal advantage over ALGE, its improvement in accuracy (+15%) is significantly better than that of Random (+6.3%) and US (+10%). Our experimental results show that cohesiveness and connectedness are useful measures in the design of an active learner.

7. CONCLUSIONS

In this paper we studied transductive classification of objects in a heterogenous information network. Through a cross-sectional study and a longitudinal study, we found that transductive classifiers give very similar performance for a given HIN classification task, but the performance of a classifier varies widely across different tasks. We proposed to study the structural properties of HINs in order to understand the intrinsic factors that determine the success of transductive classification. Through analysis, we conjectured that the validity of the connectivity assumption is strongly affected by two structural properties, namely, cohesiveness and connectedness. We proposed quantitative measures for these two properties and put forward a method to estimate their values. We conducted experiments to evaluate the reliability of the estimation. We showed how these estimates can be utilized to build a black-box tester that provides recommendation on whether an HIN classification task should be tackled by transductive classification. Furthermore, we designed an active learning algorithm ALCC, which is based on the estimated values. We conducted case studies to show that the black-box tester gives reliable recommendations and our experiments show that ALCC outperforms other active learning algorithms.

8. ACKNOWLEDGMENTS

This research is supported by Hong Kong Research Grants Council GRF grant HKU712712E.

9. REFERENCES

[1] M. Belkin et al. Regularization and semi-supervised learning on large graphs. In *COLT*, 2004.

[2] M. Belkin et al. On manifold regularization. In *AISTATS*, 2005.

[3] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *AISTATS*, 2005.

[4] M. Ji et al. Graph regularized transductive classification on heterogeneous information networks. In *ECML/PKDD*, 2010.

[5] M. Ji et al. Ranking-based classification of heterogeneous information networks. In *KDD*, 2011.

[6] X. Kong et al. Meta path-based collective classification in heterogeneous information networks. In *CIKM*, 2012.

[7] Q. Lu and L. Getoor. Link-based classification. In *ICML*, 2003.

[8] C. Luo et al. Hetpathmine: A novel transductive classification algorithm on heterogeneous information networks. In *ECIR*, 2014.

[9] S. A. Macskassy and F. Provost. A simple relational classifier. Technical report, DTIC Document, 2003.

[10] S. A. Macskassy and F. J. Provost. Classification in networked data: A toolkit and a univariate case study. *JMLR*, 2007.

[11] J. Neville and D. Jensen. Relational dependency networks. *JMLR*, 2007.

[12] Y. Sun et al. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD*, 2009.

[13] Y. Sun et al. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *VLDB*, 2011.

[14] Y. Sun et al. Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. In *KDD*, 2012.

[15] B. Taskar et al. Probabilistic classification and clustering in relational data. In *IJCAI*, 2001.

[16] B. Taskar et al. Discriminative probabilistic models for relational data. In *UAI*, 2002.

[17] C. Wan et al. Classification with active learning and meta-paths in heterogeneous information networks. In *CIKM*, 2015.

[18] M. Wan et al. Graph regularized meta-path based transductive regression in heterogeneous information network. In *SDM*, 2015.

[19] M. Wu and B. Schölkopf. Transductive classification via local learning regularization. In *AISTATS*, 2007.

[20] Z. Yin et al. Exploring social tagging graph for web object classification. In *KDD*, 2009.

[21] X. Yu et al. User guided entity similarity search using meta-path selection in heterogeneous information networks. In *CIKM*, 2012.

[22] X. Yu et al. Personalized entity recommendation: a heterogeneous information network approach. In *WSDM*, 2014.

[23] J. Zhang et al. Meta-path based multi-network collective link prediction. In *KDD*, 2014.

[24] D. Zhou et al. Learning with local and global consistency. In *NIPS*, 2004.

[25] Y. Zhou and L. Liu. Activity-edge centric multi-label classification for mining heterogeneous information networks. In *KDD*, 2014.

[26] X. Zhu et al. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.