Graph-based Classification on Heterogeneous Information Networks

Ming Ji, Yizhou Sun, Marina Danilevsky and Jiawei Han

Abstract—A heterogeneous information network is a network composed of multiple types of objects and links. Recently, it has been recognized that strongly-typed heterogeneous information networks are prevalent in the real world. Sometimes, label information is available for part of the objects. Learning from such labeled and unlabeled data via transductive classification can lead to good knowledge extraction of the hidden network structure. However, although classification on homogeneous networks has been studied over decades, classification on heterogeneous networks has not been explored until recently.
In this paper, we consider the transductive classification problem on heterogeneous networked data which share a common topic. Only part of the objects in the given network are labeled, and we aim to predict labels for all types of the remaining objects. A novel graph-based regularization framework, GNetClass, is proposed to model the link structure in information networks with arbitrary network schema and number of object/link types. Specifically, we explicitly respect the type differences by preserving consistency over each relation graph corresponding to each type of links separately. Efficient computational schemes are then introduced to solve the corresponding optimization problem. Experiments on the DBLP data set show that our algorithm significantly improves the classification accuracy over existing state-of-the-art methods.

Index Terms—Heterogeneous Information Networks, Classification, Semi-supervised Learning, Regularization.

✦

1 INTRODUCTION

INFORMATION networks, composed of large numbers of data objects linking to each other, are ubiquitous in real life. Examples include co-author networks and paper citation networks extracted from bibliographic data, and webpage networks interconnected by hyperlinks in the World Wide Web. Extracting knowledge from such gigantic sets of networked data has recently attracted substantial interest [10] [14] [15] [18]. Sometimes, label information is available for part of the data objects. Learning from labeled and unlabeled data is often called semi-supervised learning [21] [20] [3], which aims to classify the unlabeled data based on known information. Classification can help discover the hidden structure of the information network, and give deep insight into understanding different roles played by each object. In fact, applications like research community discovery, fraud detection and product recommendation can all be cast as a classification problem [10] [14]. Generally, classification can be categorized into two groups: (1) Transductive classification [9] [10] [21] [20] [18]: to predict labels for the given unlabeled data; and (2) Inductive classification [8] [14] [11] [16] [3]: to construct a decision function in the whole data space. In this paper, we focus on transductive classification, which is a common scenario in networked data.

Current studies about transductive classification on

• *Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA. E-mail: {mingji1, sun22, danilev1, hanj}@illinois.edu*

networked data [8] [9] [10] [14] mainly focus on homogeneous information networks, i.e., networks composed of a single type of objects, as mentioned above. But in real life, there could be multiple types of objects which form heterogeneous information networks. Beyond co-author networks and citation networks, the bibliographic data naturally forms a network among papers, authors, conferences, terms, etc. E-commerce systems can also be viewed as networks linking sellers, customers, items and tags. It has been recognized that heterogeneous information networks, where interconnected links can occur between any two types of objects, are prevalent.

Example 1. Bibliographic Information Network. A bibliographic information network generally contains four types of data objects: *papers*, *authors*, *venues* (conferences and journals) and *terms*. *Papers* and *authors* are linked by the relation of "written by" and "write". *Papers* and *venues* are linked by the relation of "published in" and "publish". *Papers* and *terms* are linked by the relation of "contain" and "contained in". ∎

As a natural generalization of classification on homogeneous networked data, we consider the problem of classifying heterogeneous networked data into classes, each of which is composed of multi-typed data sharing a common topic. For instance, a research community in a bibliographic information network contains not only authors, but also papers, venues and terms all belonging to the same research area. Other examples include movie networks in which movies, directors, actors and keywords relate to the same genre, and E-commerce networks where sellers, customers, items and tags belong to the same shop-

ping category.

The general problem of classification has been well studied in literature. Transductive classification on strongly-typed heterogeneous information networks, however, is much more challenging due to the characteristics of data, which are listed as follows:

1. *Complexity of the network structure.* Heterogeneous information networks contain multiple types of objects and links, and links can exist between any two types of objects. When dealing with such multi-typed network structure, one common solution is to transform it into a homogenous network and apply traditional classification methods [10] [14]. However, this simple transformation has several drawbacks. For instance, suppose we want to classify *papers* into different research areas. Existing methods would most likely extract a *citation* network from the whole bibliographic network. Then some valuable discriminative information is likely to be lost (e.g., *authors* of the paper, and *venue* the paper is published in.) What is more, it remains unclear how to optimally transform the multi-typed link information into a single-typed relation. Two *papers* can be linked either by citation or by sharing a common author. Simply adding these two types of relationship together fails to distinguish the difference in their semantic meanings. Another solution to make use of the whole network is ignoring the type differences between objects and links. Nevertheless, different types of objects naturally have different data distributions, and different types of links have different semantic meanings, therefore treating them equally is likely to be suboptimal. It has been recognized [7] [15] that while mining heterogeneous information networks, the type differences among links and objects should be respected in order to generate more meaningful results.

2. *Lack of features.* Traditional classification methods usually learn from local features or attributes of the data. However, there is no natural feature representation for all types of networked data. For example, we can use term frequency to represent a document, and use pixel values to represent an image. But it is unclear how to define meaningful feature representations for conferences and authors in a bibliographic information network. If we transform the link information into features, we will likely generate very high dimensional and sparse data as the number of objects grows up. Moreover, even if we have feature representation for some objects in a heterogeneous information network, the features of different types of objects are in difference spaces and are hardly comparable. This is another reason why traditional feature-based methods including Support Vector Machines, Naïve Bayes and logistic regression are difficult to be applied in heterogeneous information networks.

3. *Lack of labels.* Many classification approaches need a reasonable amount of training examples. However, labels are expensive in many real applications. In a heterogeneous information network, we may even not be able to have a fully labeled subset of all types of objects for training. Label information for some types of objects are easy to obtain while labels for some other types are not. Therefore, a flexible transductive classifier should be able to transfer the label information among different types of objects.

In this paper, we propose a novel graph-based regularization framework to address all three challenges, which simultaneously classifies all of the non-attributed, network-only data with an arbitrary network topology and number of object/link types, just based on the label information of any type(s) of objects and the link structure. By preserving consistency over each relation graph corresponding to each type of links separately, we explicitly respect the type differences in links and objects, thus encoding the typed information in a more organized way than traditional graph-based transductive classification on homogeneous networks.

The rest of the paper is structured as follows. In Section 2, we briefly review the existing work about classification on networked data and graph-based learning. In Section 3, we formally define the problem of transductive classification on heterogeneous information networks. Our graph-based regularization framework (denoted by GNetClass) is introduced in Section 4. Section 5 provides the experimental results. Finally, we conclude this work in Section 6.

2 RELATED WORK

W E summarize various transductive classification methods in Table 1, where one dimension represents whether the data has features/attributes or not, and the other dimension represents different kinds of network structure: from non-networked data to heterogeneous networked data. Our proposed method works on heterogeneous, non-attributed network-only data, which is the most general case requiring the least amount of information.

Classifying networked data has received substantial attention in recent years. One key difference between traditional classification and classifying networked data is that data points are not independent, but are closely correlated. So The central idea is to infer the class label from the network structure together with local attributes, if there are any. When classifying webpages or documents, local text features and the link information can be combined by using Naïve Bayes [4], logistic regression [8], graph regularization [19],

TABLE 1
Summary of related work about transductive classification

	Non-networked data	Homogenous networked data	Heterogeneous networked data
Attributed data	SVM, Graph-based learning, etc.	Statistical Relational Learning (Relational Dependency Networks, Relational Markov Networks, etc.)	
Non-attributed data	/	Network-only Link-based classifier, Relational Neighbor, etc.	*GNetClass*

etc. These methods share the same assumption that the network is homogeneous. Relational dependency networks [11] respect the type differences among relational data when learning the dependency structure by building a conditional model for each variable of interest, but still rely on local features just like other relational learning methods do. Moreover, statistical relational learning usually requires a fully labeled data set for training, which might be difficult to obtain in real applications.

Macskassy et al. [9] propose a relational neighbor classifier on network-only data. Through iteratively classifying an object by the majority class of its neighbors, this method performs very well compared to more complex models including Probabilistic Relational Models [6] [17], Relational Probability Trees [12] and Relational Bayesian Classifiers [13]. Macskassy et al. [10] further analyze various configurations in a classification framework comprised of a local classifier, a relational classifier and a collective inference procedure. They emphasize the importance of homogeneousness in within-network classification.

Recently, there has been a surge of interest in mining heterogeneous information networks [2] [7] [1], and it has been proposed that the type differences should be respected in order to generated more meaningful results. NetClus [15] uses a ranking-clustering mutual enhancement method to generate clusters composed of multi-typed objects. However, clustering does not effectively make use of prior knowledge when it is available. Yin et al. [18] explore social tagging graphs for heterogeneous web object classification. They construct a bipartite graph between tags and web objects to boost classification performance. Nevertheless, their method is confined to the specific network schema between tags and web data, thus not allowing arbitrary link structure and cannot be applied to general heterogeneous information networks.

Meanwhile, graph-based learning has enjoyed long-lasting popularity in transductive classification. Most of the methods construct an affinity graph over both the labeled and unlabeled examples based on local features to encode the similarity between instances. Then they design a learner which preserves the smoothness and consistency over the geometrical structure of the data. Zhu et al. [21] formulate the problem using a Gaussian random field model defined with respect to the graph. Zhou et al. [20]

propose to let each point iteratively spread its label information to neighbors so as to ensure both local and global consistency. Belkin et al. [3] design a framework for data-dependent regularization that exploits the geometry of the probability distribution on the labeled and unlabeled data. When local features are not available in information networks, graph-based methods can sometimes use the inherent network structure to play the role of the affinity graph. However, traditional graph-based learning mainly works on homogeneous graphs covering all the examples as a whole, and thus cannot distinguish the different semantic meaning of multi-typed links and objects very well. In this paper, we naturally extend the graph-based learning framework to fit the special requirements of heterogeneous networked data.

3 PROBLEM DEFINITION

IN this section, we introduce several related concepts and notations related to transductive classification on heterogeneous information networks, and then formally define the problem.

Definition 1. Heterogeneous information network. Given m types of data objects, denoted by $\mathcal{X}_1 = \{x_{11}, \ldots, x_{1n_1}\}$, $\ldots$, $\mathcal{X}_m = \{x_{m1}, \ldots, x_{mn_m}\}$, a graph $G = \langle V, E, W \rangle$ is called a heterogeneous information network if $V = \bigcup_{i=1}^{m} \mathcal{X}_i$ and $m \geq 2$, E is the set of links between any two data objects of V, and W is the set of weight values on the links. When $m = 1$, G reduces to a homogeneous information network.

Definition 2. Class. Given a heterogeneous information network $G = \langle V, E, W \rangle$, $V = \bigcup_{i=1}^{m} \mathcal{X}_i$, a class is defined as $G' = \langle V', E', W' \rangle$, where $V' \subseteq V$, $E' \subseteq E$. $\forall e = \langle x_{ip}, x_{jq} \rangle \in E'$, $W'_{x_{ip}x_{jq}} = W_{x_{ip}x_{jq}}$. Note here, V' also consists of multiple types of objects from $\mathcal{X}_1$ to $\mathcal{X}_m$.

Definition 2 follows [15]. Notice that a class in a heterogeneous information network is actually a sub-network containing multi-typed objects that are closely related to each other. Now our problem can be formalized as follows.

Definition 3. Transductive classification on heterogeneous information networks. Given a heterogeneous information network $G = \langle V, E, W \rangle$, a subset of data objects $V' \subseteq V = \bigcup_{i=1}^{m} \mathcal{X}_i$ which are labeled with values $\mathcal{Y}$ denoting which class each object belongs to, predict the class labels for all the unlabeled objects $V - V'$.

Fig. 1. Knowledge propagation in a bibliographic information network
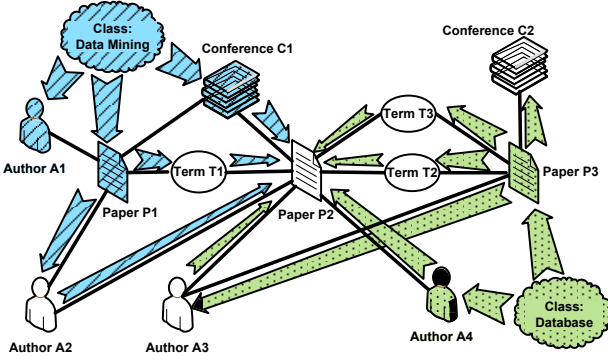
We design a set of one-versus-all soft classifiers in the multi-class classification task. Suppose the number of classes is K. For any object type $\mathcal{X}_i$, $i \in \{1, \ldots, m\}$, we try to compute a class indicator matrix $\mathbf{F}_i = [\boldsymbol{f}_i^{(1)}, \ldots, \boldsymbol{f}_i^{(K)}] \in \mathbb{R}^{n_i \times K}$, where each $\boldsymbol{f}_i^{(k)} = [f_{i1}^{(k)}, \ldots, f_{in_i}^{(k)}]^T$ measures the confidence that each object $x_{ip} \in \mathcal{X}_i$ belongs to class k. Then we can assign the p-th object in type $\mathcal{X}_i$ to class c_{ip} by finding the maximum value in the p-th row of $\mathbf{F}_i$: $c_{ip} = \arg\max_{1 \le k \le K} f_{ip}^{(k)}$.

In a heterogeneous information network, a relation graph $\mathcal{G}_{ij}$ can be built corresponding to each type of link relationship between two types of data objects $\mathcal{X}_i$ and $\mathcal{X}_j$, $i, j \in \{1, \ldots, m\}$. Note that it is possible for $i = j$. Let $\mathbf{R}_{ij}$ be an $n_i \times n_j$ relation matrix corresponding to graph $\mathcal{G}_{ij}$. The element at the p-th row and q-th column of $\mathbf{R}_{ij}$ is denoted as $R_{ij,pq}$, representing the weight on link $\langle x_{ip}, x_{jq} \rangle$. There are many ways to define the weights on the links, which can also incorporate domain knowledge. A simple definition is as follows:

$$R_{ij,pq} = \begin{cases} 1 & \text{if data objects } x_{ip} \text{ and } x_{jq} \text{ are linked} \\ 0 & \text{otherwise.} \end{cases}$$

Here we consider undirected graphs such that $\mathbf{R}_{ij} = \mathbf{R}_{ji}^T$.

In order to encode label information, we basically set a vector $\mathbf{y}_i^{(k)} = [y_{i1}^{(k)}, \ldots, y_{in_i}^{(k)}]^T \in \mathbb{R}^{n_i}$ for each data object type $\mathcal{X}_i$ such that:

$$y_{ip}^{(k)} = \begin{cases} 1 & \text{if } x_{ip} \text{ is labeled to the } k\text{-th class} \\ 0 & \text{otherwise.} \end{cases}$$

Then for each class $k \in \{1, \ldots, K\}$, our goal is to infer a set of $\boldsymbol{f}_i^{(k)}$ from $\mathbf{R}_{ij}$ and $\mathbf{y}_i^{(k)}$, $i, j \in \{1, \ldots, m\}$.

4 GRAPH-BASED REGULARIZATION FRAMEWORK

IN this section, we begin by describing the intuition of our method. Then we formulate the problem using a graph-based regularization framework. Finally, efficient computational schemes are proposed to solve the optimization problem.

4.1 Intuition

Consider a simple bibliographic information network in Figure 1. Four types of objects (*paper*, *author*, *conference* and *term*) are interconnected by multi-typed links (denoted by solid black lines) as described in Example 1. Suppose we want to classify them into research communities. Labeled objects are shaded, whereas the labels of unshaded objects are unknown. Given prior knowledge that author A1, paper P1 and conference C1 belong to the area of data mining, it is easy to infer that author A2 who *wrote* paper P1, and term T1 which is *contained* in P1, are both highly related to data mining. Similarly, author A3, conference C2, and terms T2 and T3 are likely to belong to the area of database, since they link directly to a database paper P3. For paper P2, things become more complicated because it is linked with both labeled and unlabeled objects. The confidence of belonging to a certain class may be transferred not only from labeled objects (conference C1 and author A4), but also from unlabeled ones (authors A2 and A3, terms T1, T2 and T3). The classification process can be intuitively viewed as a process of knowledge propagation throughout the network as shown in Figure 1, where the thick shaded arrows indicate possible knowledge flow. The more links between an object x and other objects of class k, the higher the confidence that x belongs to class k. Accordingly, labeled objects serve as the source of prior knowledge. Although this intuition is similar to [9] and [20], the interconnected relationship and the form of prior knowledge in heterogeneous information networks are more complex due to the typed information. Knowledge propagation through different types of links contains different semantic meaning, and thus should be treated separately.

In this way, our framework is based on the consistency assumption that the class assignments of two linked objects are likely to be similar. And the class prediction on labeled objects should be similar to their pre-assigned labels. In order to respect the link type differences, we ensure that such consistency is preserved over each relation graph corresponding to each type of links separately. We formulate our intuition as follows:

1) The estimated confidence measure of two objects x_{ip} and x_{jq} belonging to class k, $f_{ip}^{(k)}$ and $f_{jq}^{(k)}$, should be similar if x_{ip} and x_{jq} are linked together, i.e., the weight value $R_{ij,pq} > 0$.

2) The confidence estimation $\boldsymbol{f}_i^{(k)}$ should be similar to the ground truth, $\mathbf{y}_i^{(k)}$.

4.2 The Algorithm

For each relation matrix $\mathbf{R}_{ij}$, we define a diagonal matrix $\mathbf{D}_{ij}$ of size $n_i \times n_i$. The (p,p)-th element of $\mathbf{D}_{ij}$ is the sum of the p-th row of $\mathbf{R}_{ij}$. Following the above discussion, $\boldsymbol{f}_i^{(k)}$ should be as consistent as possible

with the link information and prior knowledge, so we try to minimize the following objective function:

$$
\begin{aligned}
&J(\boldsymbol{f}_1^{(k)},\ldots,\boldsymbol{f}_m^{(k)}) \\
&= \sum_{i,j=1}^{m} \lambda_{ij} \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} R_{ij,pq} \Big(\frac{1}{\sqrt{D_{ij,pp}}} f_{ip}^{(k)} - \frac{1}{\sqrt{D_{ji,qq}}} f_{jq}^{(k)} \Big)^2 \\
&\quad + \sum_{i=1}^{m} \alpha_i (\boldsymbol{f}_i^{(k)} - \mathbf{y}_i^{(k)})^T (\boldsymbol{f}_i^{(k)} - \mathbf{y}_i^{(k)}). \quad (1)
\end{aligned}
$$

where $D_{ij,pp}$ is the (p,p)-th element of $\mathbf{D}_{ij}$, and $D_{ji,qq}$ is the (q,q)-th element of $\mathbf{D}_{ji}$. The first term in the objective function (1) is the *smoothness* constraints formulating the first intuition. This term is normalized by $\sqrt{D_{ij,pp}}$ and $\sqrt{D_{ji,qq}}$ in order to reduce the impact of popularity of nodes. In other words, we can, to some extent, suppress popular nodes from dominating the confidence estimations. The normalization technique is adopted in traditional graph-based learning with the effectiveness well proved [20]. The second term minimizes the difference between the prediction results and the labels, reflecting the second intuition.

The trade-off among different terms is controlled by regularization parameters λ_{ij} and α_i, where $0 \le \lambda_{ij} < 1$, $0 < \alpha_i < 1$. For $\forall i,j \in \{1,\ldots,m\}$, $\lambda_{ij} > 0$ indicates that object types $\mathcal{X}_i$ and $\mathcal{X}_j$ are linked together and this relationship is taken into consideration. The larger λ_{ij}, the more value is placed on the relationship between object types $\mathcal{X}_i$ and $\mathcal{X}_j$. For example, in a bibliographic information network, if a user believes that the links between *authors* and *papers* are more trustworthy and influential than the links between *conferences* and *papers*, then the λ_{ij} corresponding to the *author-paper* relationship should be set larger than that of *conference-paper*, and the classification results will rely more on the *author-paper* relationship. Similarly, the value of α_i, to some extent, measures how much the user trusts the given prior knowledge of object type $\mathcal{X}_i$. The parameters thus provide more flexibility to incorporate user preference. But we will show in Section 5 that the parameter setting will not influence the performance of our algorithm dramatically.

To facilitate algorithm derivation, we define the normalized form of $\mathbf{R}_{ij}$:

$$
\mathbf{S}_{ij} = \mathbf{D}_{ij}^{(-1/2)} \mathbf{R}_{ij} \mathbf{D}_{ji}^{(-1/2)}, i,j \in \{1,\ldots,m\} \quad (2)
$$

With simple algebraic formulations, the first term

of (1) can be rewritten as:

$$
\begin{aligned}
&\sum_{i,j=1}^{m} \lambda_{ij} \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} R_{ij,pq} \Big(\frac{f_{ip}^{(k)}}{\sqrt{D_{ij,pp}}} - \frac{f_{jq}^{(k)}}{\sqrt{D_{ji,qq}}} \Big)^2 \\
&= \sum_{i,j=1}^{m} \lambda_{ij} \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} R_{ij,pq} \Big(\frac{(f_{ip}^{(k)})^2}{D_{ij,pp}} - 2 \frac{f_{ip}^{(k)} f_{jq}^{(k)}}{\sqrt{D_{ij,pp} D_{ji,qq}}} \\
&\quad + \frac{(f_{jq}^{(k)})^2}{D_{ji,qq}} \Big) \\
&= \sum_{i,j=1}^{m} \lambda_{ij} \Big(\sum_{p=1}^{n_i} (f_{ip}^{(k)})^2 + \sum_{q=1}^{n_j} (f_{jq}^{(k)})^2 \\
&\quad - 2 \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} (f_{ip}^{(k)} S_{ij,pq} f_{jq}^{(k)}) \Big) \\
&= \sum_{i,j=1}^{m} \lambda_{ij} \big((\boldsymbol{f}_i^{(k)})^T \boldsymbol{f}_i^{(k)} + (\boldsymbol{f}_j^{(k)})^T \boldsymbol{f}_j^{(k)} - 2(\boldsymbol{f}_i^{(k)})^T \mathbf{S}_{ij} \boldsymbol{f}_j^{(k)} \big)
\end{aligned}
$$

Then we can rewrite (1) in the following form:

$$
\begin{aligned}
&J(\boldsymbol{f}_1^{(k)},\ldots,\boldsymbol{f}_m^{(k)}) \\
&= \sum_{i,j=1}^{m} \lambda_{ij} \big((\boldsymbol{f}_i^{(k)})^T \boldsymbol{f}_i^{(k)} + (\boldsymbol{f}_j^{(k)})^T \boldsymbol{f}_j^{(k)} - 2(\boldsymbol{f}_i^{(k)})^T \mathbf{S}_{ij} \boldsymbol{f}_j^{(k)} \big) \\
&\quad + \sum_{i=1}^{m} \alpha_i (\boldsymbol{f}_i^{(k)} - \mathbf{y}_i^{(k)})^T (\boldsymbol{f}_i^{(k)} - \mathbf{y}_i^{(k)}) \quad (3)
\end{aligned}
$$

4.2.1 Connection to homogeneous graph-based learning.

Here we first show that the homogenous version of our algorithm is equivalent to the graph-based learning method [20]. Then we show the connection and difference between our algorithm and [20] on heterogeneous information networks.

We first define $\mathbf{L}_{ii} = \mathbf{I}_i - \mathbf{S}_{ii}$, where $\mathbf{I}_i$ is the identity matrix of size $n_i \times n_i$. Note that $\mathbf{L}_{ii}$ is the *normalized graph Laplacian* [5] of the homogeneous sub-network on object type $\mathcal{X}_i$.

Lemma 1. *In homogeneous information networks, the objective function (3) reduces to:*

$$
J(\boldsymbol{f}_1^{(k)}) = 2\lambda_{11}(\boldsymbol{f}_1^{(k)})^T \mathbf{L}_{11} \boldsymbol{f}_1^{(k)} + \alpha_1 (\boldsymbol{f}_1^{(k)} - \mathbf{y}_1^{(k)})^T (\boldsymbol{f}_1^{(k)} - \mathbf{y}_1^{(k)}) \quad \blacksquare
$$

The proof can be done by simply setting $m = 1$ in function (3). It is easy to see that the homogeneous version of our algorithm is equivalent to the objective function of [20].

When the information network is heterogeneous, we can consider all types of objects as a whole set. We define:

$$
\begin{aligned}
\boldsymbol{f}^{(k)} &= [(\boldsymbol{f}_1^{(k)})^T, \ldots, (\boldsymbol{f}_m^{(k)})^T]^T \\
\mathbf{y}^{(k)} &= [(\mathbf{y}_1^{(k)})^T, \ldots, (\mathbf{y}_m^{(k)})^T]^T \\
\boldsymbol{\alpha}_i &= \alpha_i \mathbf{1}_{n_i}, i = 1,\ldots,m \\
\boldsymbol{\alpha} &= \mathrm{diag}\{[\boldsymbol{\alpha}_1^T, \ldots, \boldsymbol{\alpha}_m^T]\}
\end{aligned}
$$

where $\mathbf{1}_{n_i}$ is an n_i-dimensional column vector of all ones. We further construct a matrix corresponding to

each type of relationship between two different object types $\mathcal{X}_i$ and $\mathcal{X}_j$ as follows:

$$\mathbf{L}_{ij} = \begin{bmatrix} \mathbf{I}_i & -\mathbf{S}_{ij} \\ -\mathbf{S}_{ji} & \mathbf{I}_j \end{bmatrix}, \text{ where } i \neq j$$

Suppose $\sum_{i=1}^m n_i = n$, let $\mathbf{H}_{ij}$ be the $n \times n$ symmetric matrix where each row/column corresponds to an object, with the order same as that in $\boldsymbol{f}^{(k)}$. The elements of $\mathbf{H}_{ij}$ at rows and columns corresponding to object types $\mathcal{X}_i$ and $\mathcal{X}_j$ are equal to $\mathbf{L}_{ij}$, and all the other elements are 0. This also holds for $i = j$.

Lemma 2. *On heterogeneous information networks, the objective function (3) is equivalent to the following:*

$$J(\boldsymbol{f}_1^{(k)}, \ldots, \boldsymbol{f}_m^{(k)})$$
$$= (\boldsymbol{f}^{(k)})^T \mathbf{H} \boldsymbol{f}^{(k)} + (\boldsymbol{f}^{(k)} - \mathbf{y}^{(k)})^T \boldsymbol{\alpha} (\boldsymbol{f}^{(k)} - \mathbf{y}^{(k)}) \text{ (4)}$$

where $\mathbf{H} = \sum_{i \neq j} \lambda_{ij} \mathbf{H}_{ij} + 2\sum_{i=1}^m \lambda_{ii} \mathbf{H}_{ii}$. ∎

Proof: We take different terms in objective function (3) separately. Recall that

$$\mathbf{R}_{ij} = \mathbf{R}_{ji}^T \Longrightarrow \mathbf{S}_{ij} = \mathbf{S}_{ji}^T$$

and

$$(\boldsymbol{f}_i^{(k)})^T \mathbf{S}_{ij} \boldsymbol{f}_j^{(k)} = (\boldsymbol{f}_j^{(k)})^T \mathbf{S}_{ji} \boldsymbol{f}_i^{(k)}$$

When $i \neq j$, we have:

$$\lambda_{ij}\big((\boldsymbol{f}_i^{(k)})^T \boldsymbol{f}_i^{(k)} + (\boldsymbol{f}_j^{(k)})^T \boldsymbol{f}_j^{(k)} - 2(\boldsymbol{f}_i^{(k)})^T \mathbf{S}_{ij} \boldsymbol{f}_j^{(k)}\big)$$
$$= \lambda_{ij} \begin{bmatrix} (\boldsymbol{f}_i^{(k)})^T & (\boldsymbol{f}_j^{(k)})^T \end{bmatrix} \begin{bmatrix} \mathbf{I}_i & -\mathbf{S}_{ij} \\ -\mathbf{S}_{ji} & \mathbf{I}_j \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_i^{(k)} \\ \boldsymbol{f}_j^{(k)} \end{bmatrix}$$
$$= \lambda_{ij} \begin{bmatrix} (\boldsymbol{f}_i^{(k)})^T & (\boldsymbol{f}_j^{(k)})^T \end{bmatrix} \mathbf{L}_{ij} \begin{bmatrix} \boldsymbol{f}_i^{(k)} \\ \boldsymbol{f}_j^{(k)} \end{bmatrix}$$
$$= \lambda_{ij} (\boldsymbol{f}^{(k)})^T \mathbf{H}_{ij} \boldsymbol{f}^{(k)}$$

When $i = j$, we have:

$$\lambda_{ij}\big((\boldsymbol{f}_i^{(k)})^T \boldsymbol{f}_i^{(k)} + (\boldsymbol{f}_j^{(k)})^T \boldsymbol{f}_j^{(k)} - 2(\boldsymbol{f}_i^{(k)})^T \mathbf{S}_{ij} \boldsymbol{f}_j^{(k)}\big)$$
$$= \lambda_{ii}\big(2(\boldsymbol{f}_i^{(k)})^T \boldsymbol{f}_i^{(k)} - 2(\boldsymbol{f}_i^{(k)})^T \mathbf{S}_{ii} \boldsymbol{f}_i^{(k)}\big)$$
$$= 2\lambda_{ii}(\boldsymbol{f}_i^{(k)})^T (\mathbf{I}_i - \mathbf{S}_{ii}) \boldsymbol{f}_i^{(k)}$$
$$= 2\lambda_{ii}(\boldsymbol{f}_i^{(k)})^T \mathbf{L}_{ii} \boldsymbol{f}_i^{(k)}$$
$$= 2\lambda_{ii}(\boldsymbol{f}^{(k)})^T \mathbf{H}_{ii} \boldsymbol{f}^{(k)}$$

And it is easy to see that:

$$\sum_{i=1}^m \alpha_i (\boldsymbol{f}_i^{(k)} - \mathbf{y}_i^{(k)})^T (\boldsymbol{f}_i^{(k)} - \mathbf{y}_i^{(k)})$$
$$= \big((\boldsymbol{f}^{(k)}) - \mathbf{y}^{(k)}\big)^T \boldsymbol{\alpha} \big((\boldsymbol{f}^{(k)}) - \mathbf{y}^{(k)}\big)$$

Taking the summation of all the terms together:

$$J(\boldsymbol{f}_1^{(k)}, \ldots, \boldsymbol{f}_m^{(k)})$$
$$= \sum_{i \neq j} \lambda_{ij} (\boldsymbol{f}^{(k)})^T \mathbf{H}_{ij} \boldsymbol{f}^{(k)} + 2\sum_{i=1}^m \lambda_{ii} (\boldsymbol{f}^{(k)})^T \mathbf{H}_{ii} \boldsymbol{f}^{(k)}$$
$$+ \big((\boldsymbol{f}^{(k)}) - \mathbf{y}^{(k)}\big)^T \boldsymbol{\alpha} \big((\boldsymbol{f}^{(k)}) - \mathbf{y}^{(k)}\big)$$
$$= (\boldsymbol{f}^{(k)})^T \mathbf{H} \boldsymbol{f}^{(k)} + (\boldsymbol{f}^{(k)} - \mathbf{y}^{(k)})^T \boldsymbol{\alpha} (\boldsymbol{f}^{(k)} - \mathbf{y}^{(k)})$$

□

Lemma 2 shows that our proposed GNetClass algorithm has a consistent form with the graph-based learning framework on homogeneous data [20], in which $\mathbf{H}$ is replaced by the *normalized graph Laplacian* $\mathbf{L}$ [5]. Moreover, we respect the different semantic meanings of the multi-typed links by applying graph regularization on each relation graph corresponding to each type of links separately rather than on the whole network. Different regularization parameters λ_{ij} also provide more flexibility in incorporating user preference on how much the relationship between object types $\mathcal{X}_i$ and $\mathcal{X}_j$ is valued among all types of relationships. However, even if all the λ_{ij} are set the same, we can see that $\mathbf{H}$ is different from the *normalized graph Laplacian* $\mathbf{L}$ [5] on the whole network as long as there is one type of objects linking to other objects via multiple types of relationships.[1]

4.2.2 Closed form solution.

It is easy to check that $\mathbf{L}_{ii}$ is positive semi-definite, and so is $\mathbf{H}_{ii}$. We now show that $\mathbf{L}_{ij}$ is also positive semi-definite.

Proof: Recall that $D_{ij,pp} = \sum_{q=1}^{n_j} R_{ij,pq}$ and $\mathbf{R}_{ij} = \mathbf{R}_{ji}^T$, we define:

$$\widehat{\mathbf{L}}_{ij} = \begin{bmatrix} \mathbf{D}_{ij} & -\mathbf{R}_{ij} \\ -\mathbf{R}_{ji} & \mathbf{D}_{ji} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{D}_{ij} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{ji} \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \mathbf{R}_{ij} \\ \mathbf{R}_{ji} & \mathbf{0} \end{bmatrix} = \widehat{\mathbf{D}} - \widehat{\mathbf{W}}$$

It can be observed that $\widehat{\mathbf{L}}_{ij}$ has the same form as the *graph Laplacian* [5], where $\widehat{\mathbf{D}}$ is a diagonal matrix whose entries are column (or row, since $\widehat{\mathbf{W}}$ is symmetric) sums of $\widehat{\mathbf{W}}$. So $\widehat{\mathbf{L}}_{ij}$ is positive semi-definite. Hence

$$\mathbf{L}_{ij} = \begin{bmatrix} \mathbf{D}_{ij} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{ji} \end{bmatrix}^{-1/2} \widehat{\mathbf{L}}_{ij} \begin{bmatrix} \mathbf{D}_{ij} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{ji} \end{bmatrix}^{-1/2}$$

is positive semi-definite. □

In this way, $\mathbf{H}_{ij}$ is positive semi-definite. We further check the Hessian matrix of the objective function (3), which is easy to derive from equation (4):

$$H\big(J(\boldsymbol{f}_1^{(k)}, \ldots, \boldsymbol{f}_m^{(k)})\big) = 2\mathbf{H} + 2\boldsymbol{\alpha}$$

$\mathbf{H}$ is the weighted summation of $\mathbf{H}_{ii}$ and $\mathbf{H}_{ij}$, which is also positive semi-definite. Since $\alpha_i > 0$ for all i, we conclude that $H\big(J(\boldsymbol{f}_1^{(k)}, \ldots, \boldsymbol{f}_m^{(k)})\big)$ is positive definite. Therefore, the objective function (3) is strictly convex. The unique global minimum is obtained by differentiating (3) with respect to each $(\boldsymbol{f}_i^{(k)})^T$:

$$\frac{\partial J}{\partial (\boldsymbol{f}_i^{(k)})^T} = \sum_{j=1, j \neq i}^m \lambda_{ij}(2\boldsymbol{f}_i^{(k)} - 2\mathbf{S}_{ij}\boldsymbol{f}_j^{(k)})$$
$$+ 4\lambda_{ii}\mathbf{L}_{ii}\boldsymbol{f}_i^{(k)} + 2\alpha_i(\boldsymbol{f}_i^{(k)} - \mathbf{y}_i^{(k)}) \quad \text{(5)}$$

1. If a network has only two types of objects $\mathcal{X}_1$ and $\mathcal{X}_2$, and only one type of relationship $\mathbf{R}_{12}$, then $\mathbf{H}$ reduces to $\lambda_{12}\mathbf{L}$.

and letting $\frac{\partial J}{\partial (f_i^{(k)})^T} = 0$ for all i.

Finally, we give the closed form solution by solving the following linear equation system:

$$
\begin{aligned}
\boldsymbol{f}_i^{(k)} \;=\; & \Big(\big(\sum_{j=1, j\neq i}^{m} \lambda_{ij} + \alpha_i \big) \mathbf{I}_i + 2\lambda_{ii}\mathbf{L}_{ii} \Big)^{-1} \\
& \times \Big(\alpha_i \mathbf{y}_i^{(k)} + \sum_{j=1, j\neq i}^{m} \lambda_{ij}\mathbf{S}_{ij}\boldsymbol{f}_j^{(k)} \Big) , \; i \in \{1, \ldots, m\}
\end{aligned}
$$

It can be proven that $\Big(\big(\sum_{j=1, j\neq i}^{m} \lambda_{ij} + \alpha_i \big) \mathbf{I}_i + 2\lambda_{ii}\mathbf{L}_{ii} \Big)$ is positive definite and invertible.

4.2.3 Iterative solution.

Though the closed form solution is obtained, sometimes the iterative solution is preferable. Based on equation (5), we derive the iterative form of our algorithm as follows:

- Step 0: For $\forall k \in \{1, \ldots, K\}$, $\forall i \in \{1, \ldots, m\}$, initialize confidence estimates $\boldsymbol{f}_i^{(k)}(0) = \mathbf{y}_i^{(k)}$ and $t = 0$.
- Step 1: Based on the current $\boldsymbol{f}_i^{(k)}(t)$, compute:

$$
\begin{aligned}
& \boldsymbol{f}_i^{(k)}(t+1) \\
& = \frac{\sum_{j=1, j\neq i}^{m} \lambda_{ij}\mathbf{S}_{ij}\boldsymbol{f}_j^{(k)}(t) + 2\lambda_{ii}\mathbf{S}_{ii}\boldsymbol{f}_i^{(k)}(t) + \alpha_i\mathbf{y}_i^{(k)}}{\sum_{j=1, j\neq i}^{m} \lambda_{ij} + 2\lambda_{ii} + \alpha_i}
\end{aligned}
$$

for $\forall k \in \{1, \ldots, K\}$, $\forall i \in \{1, \ldots, m\}$.
- Step 2: Repeat step 1 with $t = t + 1$ until convergence, i.e., until $\boldsymbol{f}_i^{(k)*} = \boldsymbol{f}_i^{(k)}(t)$ do not change much for all i.
- Step 3: For each $i \in \{1, \ldots, m\}$, assign the class label to the p-th object of type $\mathcal{X}_i$ as $c_{ip} = \arg\max_{1\leq k\leq K} f_{ip}^{(k)*}$, where $\boldsymbol{f}_i^{(k)*} = [f_{i1}^{(k)*}, \ldots, f_{in_i}^{(k)*}]^T$.

Following analysis similar to [20], the iterative algorithm can be proven to converge to the closed form solution. The iterative solution can be viewed as a natural extension of [20], where each object iteratively spreads label information to its neighbors until a global stable state is achieved. At the same time, we explicitly distinguish the semantic differences between the multi-typed links and objects by employing different normalized relation graphs corresponding to each type of links separately rather than a single graph covering all the instances.

4.3 Time complexity analysis

We analyze the computational complexity of the iterative solution here. Step 0 takes $O(K|V|)$ time for initialization, where K is the number of classes and $|V|$ the total number of objects. At each iteration of step 1, we need to process each link twice, once for the object at each end of the link. And we need $O(K|V|)$ time to incorporate label information in $\alpha_i\mathbf{y}_i^{(k)}$. So the time for each iteration is $O(K(|E|+|V|))$, where $|E|$ is the total number of links in the information network. Finally, it takes $O(K|V|)$ time to compute the class prediction result in step 3. Hence the total time complexity of the iterative algorithm is $O(NK(|E|+|V|))$, where N is the number of iterations.

The time complexity of the closed form solution is dependent on the particular network structure. We omit the analysis due to space limitation. In general, the iterative solution is more computationally efficient because it bypasses the matrix inversion operation.

After all, the classification task is done offline, where all the objects can be classified once and the results stored for future querying.

5 EXPERIMENTAL RESULTS

IN this section, we present an empirical study of the effectiveness of our graph-based regularization framework for transductive classification (denoted by GNetClass) on the real heterogeneous information network of DBLP[2]. As discussed before, we try to classify the bibliographic data into research communities, each of which contains multi-typed objects all closely related to the same area.

5.1 Data set

We extract a sub-network of the DBLP data set on four areas: database, data mining, information retrieval and artificial intelligence, which naturally form four classes. Five representative conferences in each area are picked. By selecting papers published in the 20 conferences, together with the authors of these papers and the terms that appeared in the titles of these papers, we obtain a heterogeneous information network that consists of four types of objects: *paper, conference, author* and *term*. Within that heterogeneous information network, we have three types of link relationships: *paper-conference*, *paper-author*, and *paper-term*. The final data set we used contains 14376 papers, 20 conferences, 14475 authors and 8920 terms, with a total number of 170794 links[3]. By using our GNetClass algorithm, we can label any type(s) of objects and simultaneously classify all types of objects.

For accuracy evaluation, we manually labeled 4057 authors according to their homepages, 100 papers according to the contents, and all 20 conferences according to their call for papers. In the following sections, we randomly choose a subset of labeled objects and use their label information as prior knowledge. The classification accuracy is evaluated by comparing with manually labeled results on the rest of the labeled objects. Since terms are difficult to label even manually, i.e., many terms are closely related to multiple areas, we did not evaluate the accuracy on terms here.

TABLE 2
Comparison of classification accuracy on authors (%)

# of conferences labeled per area	nLB(A-C-P-T)	wvRN(A-C-P-T)	LLGC(A-C-P-T)	GNetClass(A-C-P-T)
1	26.1	63.0	67.6	**70.1**
2	25.2	74.1	90.8	**91.4**
3	27.0	82.7	90.7	**91.5**

TABLE 3
Comparison of classification accuracy on papers (%)

# of conferences labeled per area	nLB(A-C-P-T)	wvRN(A-C-P-T)	LLGC(A-C-P-T)	GNetClass(A-C-P-T)
1	29.0	52.0	69.0	**72.0**
2	32.0	65.0	83.0	**85.0**
3	32.0	74.0	**84.0**	**84.0**

TABLE 4
Comparison of classification accuracy on conferences (%)

# of conferences labeled per area	nLB (C-C)	nLB (A-C-P-T)	wvRN (C-C)	wvRN (A-C-P-T)	LLGC (C-C)	LLGC (A-C-P-T)	GNetClass (A-C-P-T)
1	25.0	18.8	50.0	50.0	**75.0**	**75.0**	**75.0**
2	50.0	41.7	**83.3**	41.7	**83.3**	**83.3**	**83.3**
3	87.5	50.0	87.5	50.0	87.5	**100.0**	**100.0**

5.2 Algorithms for comparison

We compare GNetClass with the following state-of-the-art algorithms:

- Learning with Local and Global Consistency (LLGC) [20]
- Weighted-vote Relational Neighbor classifier (wvRN) [9] [10]
- Network-only Link-based Classification (nLB) [8] [10]

LLGC is a graph-based transductive classification algorithm, which is also the homogenous reduction of GNetClass if we use the intrinsic network structure to play the role of the affinity graph. Weighted-vote relational neighbor classifier and link-based classification are two popular classification algorithms on networked data. Since local attributes/features are not available in our problem, we use the network-only derivative of the link-based classifier (nLB). Following [10], nLB creates a feature vector for each node based on neighboring information.

Note that none of the algorithms above can be directly applied to heterogeneous information networks. In order to make all the algorithms comparable, we can transform a heterogenous information network into a homogeneous one in two ways: (1) disregard the type differences between objects and treat all of them as the same type; or (2) extract a homogeneous sub-network on one single type of objects, if that object type is partially labeled. We try both approaches in the accuracy study. The open-source implementation of NetKit-SRL[4] [10] is employed in our experiments.

For parameter setting, since the homogeneous LLGC algorithm just has one α and one λ, only the

4. http://www.research.rutgers.edu/~sofmac/NetKit.html

ratio $\frac{\alpha}{\lambda}$ matters. The $\frac{\alpha}{\lambda}$ are are set by searching the grid $\{0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$, where the best results are obtained by $\frac{\alpha}{\lambda} = 0.5$. For GNetClass, we do not treat any object/link type as particularly important here and use the same set of parameters as LLGC for a fair comparison, i.e. $\alpha_i = 0.1$, $\lambda_{ij} = 0.2$, $\forall i, j \in \{1, \ldots, m\}$.

5.3 Infer from a single type of objects

In this section, we only label the object type *conference* and do classification on *papers, conferences, authors* and *terms* simultaneously. Since prior knowledge is given on conferences, the results on authors and papers of wvRN, nLB and LLGC can only be obtained by disregarding the type differences between objects and links, denoted by (A-C-P-T). While classifying conferences, we also tried constructing a homogeneous *conference-conference* (C-C) sub-network in different ways. The best results shown are given by linking two conferences if an author publishes papers in both of them, with the weight being the number of common authors shared. Note that the transformation encodes link relationships of both *paper-conference* and *paper-author*. We show the classification accuracy on authors, papers and conferences in Tables 2, 3 and 4, respectively.

When classifying conferences, GNetClass performs slightly better than LLGC, wvRN and nLB on the homogeneous *conference-conference* sub-network (C-C). As the number of labeled conferences increases, the difference between these algorithms becomes smaller. LLGC on the whole network (A-C-P-T) performs comparably to GNetClass. One possible reason for this might be that the links connected with conferences are relatively dense and contain rich information, making it easy for algorithms to classify conferences correctly.

TABLE 5
Comparison of classification accuracy on authors (%)

$(a\%, p\%)$ of authors and papers labeled	nLB (A-A)	nLB (A-C-P-T)	wvRN (A-A)	wvRN (A-C-P-T)	LLGC (A-A)	LLGC (A-C-P-T)	GNetClass (A-C-P-T)
(0.1%, 0.1%)	25.4	26.0	40.8	34.1	41.4	61.3	**82.9**
(0.2%, 0.2%)	28.3	26.0	46.0	41.2	44.7	62.2	**83.4**
(0.3%, 0.3%)	28.4	27.4	48.6	42.5	48.8	65.7	**86.7**
(0.4%, 0.4%)	30.7	26.7	46.3	45.6	48.7	66.0	**87.2**
(0.5%, 0.5%)	29.8	27.3	49.0	51.4	50.6	68.9	**87.5**

TABLE 6
Comparison of classification accuracy on papers (%)

$(a\%, p\%)$ of authors and papers labeled	nLB (P-P)	nLB (A-C-P-T)	wvRN (P-P)	wvRN (A-C-P-T)	LLGC (P-P)	LLGC (A-C-P-T)	GNetClass (A-C-P-T)
(0.1%, 0.1%)	49.8	31.5	62.0	42.0	67.2	62.7	**79.2**
(0.2%, 0.2%)	73.1	40.3	71.7	49.7	72.8	65.5	**83.5**
(0.3%, 0.3%)	77.9	35.4	77.9	54.3	76.8	66.6	**83.2**
(0.4%, 0.4%)	79.1	38.6	78.1	54.4	77.9	70.5	**83.7**
(0.5%, 0.5%)	80.7	39.3	77.9	53.5	79.0	73.5	**84.1**

TABLE 7
Comparison of classification accuracy on conferences (%)

$(a\%, p\%)$ of authors and papers labeled	nLB (A-C-P-T)	wvRN (A-C-P-T)	LLGC (A-C-P-T)	GNetClass (A-C-P-T)
(0.1%, 0.1%)	25.5	43.5	79.0	**81.0**
(0.2%, 0.2%)	22.5	56.0	83.5	**85.0**
(0.3%, 0.3%)	25.0	59.0	**87.0**	**87.0**
(0.4%, 0.4%)	25.0	57.0	86.5	**89.5**
(0.5%, 0.5%)	25.0	68.0	90.0	**94.0**

When the entire heterogeneous information network (A-C-P-T) is taken into consideration, the task actually becomes more challenging, because the total number of objects rises to $14376(paper) + 20(conf) + 14475(author) + 8920(term) = 37791$, out of which at most $4 \times 3(conf)/37791 \times 100\% = 0.03\%$ objects are labeled. Similar results have been reported [10] that when the percentage of labeled objects is less than 20%, the classification accuracy can drop below random guess (here 25%). In this way, nLB performs less well due to the lack of labels. And increasing the label ratio from 0.01% to 0.03% does not really improve the prediction accuracy of nLB. Overall, GNetClass performs the best when classifying authors and papers, which are more difficult tasks due to the sparse links. Although the parameters for different types of objects and links are set the same, i.e., we treat all the types as equally important, and labels are given on only one type of objects, GNetClass still outperforms its homogeneous reduction (LLGC) by transferring the label information to other types of objects in a more effective way.

5.4 Infer from multiple types of objects

In a real heterogeneous information network, a user may have prior knowledge for multiple types of objects rather than a single type. Therefore, here we label both *authors* and *papers* to test the perfor-

mance of learning from multi-typed labels. In order to address the label scarcity problem, we randomly choose $(a\%, p\%) = [(0.1\%, 0.1\%), (0.2\%, 0.2\%), \ldots, (0.5\%, 0.5\%)]$ of authors and papers, and use their label information for transductive classification. For each given $(a\%, p\%)$, we average the results over 10 random selections. Note that the very small percentage of labeled objects here are likely to be disconnected, so we may even not be able to extract a fully labeled sub-network for training, making many state-of-the-art algorithms inapplicable.

As label information is given on authors and papers, the results on conferences of wvRN, nLB and LLGC can only be obtained by disregarding the type differences between objects and links, denoted by (A-C-P-T). While classifying authors and papers, we also tried constructing homogeneous *author-author* (A-A) and *paper-paper* (P-P) sub-networks in different ways, where the best results presented for authors are given by the *co-author* network, and the best results for papers are generated by linking two papers if they are published in the same conference. We show the classification accuracy on authors, papers and conferences in Tables 5, 6 and 7, respectively.

When classifying authors and papers, it is interesting to notice that the performances of wvRN and nLB on the *author-author* and *paper-paper* sub-networks are better than working on the whole heterogeneous information network, verifying the importance of ho-

Fig. 2. Model Selection when (0.5%, 0.5%) of authors and papers are labeled

(a) Varying α_a (b) Varying α_p (c) Varying λ_{pa}
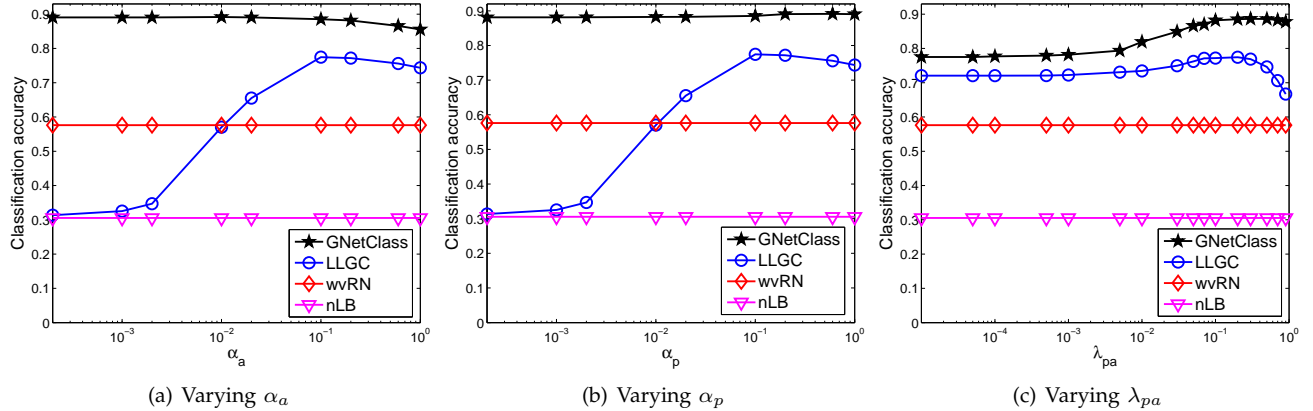
mogeneousness in such homogeneous relational classifiers. However, the transformation from the original heterogeneous network to the homogeneous sub-network causes some information loss, as discussed before. And only one type of label information can be used in the homogeneous sub-network, even if the prior knowledge of another type of objects is available.

When the entire heterogeneous information network (A-C-P-T) is taken into consideration, the task actually becomes more challenging, since the percentage of labeled objects is at most $(14475\ (authors) + 14376\ (papers)) \times 0.5\%/37791 = 0.4\%$. Similar results have been reported [10] that when the percentage of labeled objects is less than 20%, the classification accuracy can drop below random guess (here 25%). Therefore, wvRN and nLB perform less well due to the lack of labels. And increasing the label ratio from 0.1% to 0.5% does not make a big difference in improving the accuracy of nLB.

Overall, GNetClass performs the best on all types of objects via learning from labels on both authors and papers. Even though the parameters for all types of objects and links are set the same, GNetClass still outperforms its homogeneous reduction, LLGC, by preserving consistency on each subgraph corresponding to each type of links separately and minimizing the aggregated error, thus modeling the heterogenous network structure in a more organized way.

5.5 Model selection

The α_i and λ_{ij} are essential parameters in GNetClass which control the relative importance of different terms. We empirically set all the α_i as 0.1, and all the λ_{ij} as 0.2 in the previous experiments. In this subsection, we try to study the impact of parameters on the performance of GNetClass when labels are given on authors and papers. In that scenario, the α_i associated with authors (denoted by α_a) and papers (denoted by α_p), as well as the λ_{ij} associated with the *author-paper*

relationship (denoted by λ_{pa}) are empirically more important than other parameters. So we fix all the other parameters and let α_a, α_p and λ_{pa} vary. We also change the α and λ in LLGC accordingly. Figure 2 shows the average classification accuracy on three types of objects (author, paper, conference) as a function of the parameters, with $(a\%, p\%) = (0.5\%, 0.5\%)$ authors and papers labeled.

It can be observed that over a large range of parameters, GNetClass achieves significantly better performance than all the other algorithms, including its homogeneous reduction, LLGC, with the parameters varying the same way. It is interesting to note that the accuracy curve of α_a is different from that of α_p, indicating that authors and papers do play different roles in the classification process. According to the figure, it is better to set α_a smaller than other $\alpha_i (= 0.1)$. This is because many authors may work in several areas concurrently, therefore assigning an author strictly to one major area as prior knowledge is not very accurate. So α_a can be decreased a little to model the distrust on the labeled authors. In contrast, increasing α_p a little can improve the performance, as the area of one paper is usually accurate. Setting λ_{pa} larger than other $\lambda_{ij} (= 0.2)$ also improves the accuracy, as increasing λ_{pa} can enhance the knowledge propagation between the two types of labeled data, which is beneficial. Overall, the parameter selection will not critically affect the classification accuracy in GNetClass.

6 CONCLUSIONS

IN this paper, we develop a novel graph-based regularization framework to address the transductive classification problem on heterogeneous information networks. We propose that different types of objects and links should be treated separately due to different semantic meanings, which is then proved by both theory and practice. By applying graph regularization to preserve consistency over each relation graph

corresponding to each type of links separately and minimizing the aggregated error, we make full use of the multi-typed link information to predict the class label for each object. In this way, our framework can be generally applied to heterogeneous information networks with an arbitrary schema consisting of a number of object/link types. Experiments on the real DBLP data set illustrate the superiority of our method over existing algorithms. Given a limited amount of labels on different types of objects, we can do qualified classification on all types of objects simultaneously. This empirically shows that knowledge can propagate through links effectively across the whole interconnected network.

The presented framework classifies the unlabeled data by labeling some randomly selected objects. However, the quality of labels can significantly influence the classification results, as observed in many past studies. In the future, we plan to automatically detect the most informative objects, which can lead to better classification quality if they are labeled. Objects that will potentially have high ranks or lie in the centrality of sub-networks might be good candidates.

REFERENCES

[1] A. Banerjee, S. Basu, and S. Merugu. Multi-way clustering on relation graphs. In *SDM '07*, 2007.

[2] R. Bekkerman, R. El-Yaniv, and A. McCallum. Multi-way distributional clustering via pairwise interactions. In *ICML '05*, pages 41–48, 2005.

[3] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from examples. *J. Mach. Learn. Res.*, 7:2399–2434, 2006.

[4] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *SIGMOD '98*, pages 307–318. ACM, 1998.

[5] F. R. K. Chung. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics*. AMS, 1997.

[6] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *IJCAI'99*.

[7] B. Long, Z. M. Zhang, and X. W. P. S. Yu. Spectral clustering for multi-type relational data. In *ICML '06*, pages 585–592, 2006.

[8] Q. Lu and L. Getoor. Link-based classification. In *Proceedings of ICML '03*, 2003.

[9] S. A. Macskassy and F. Provost. A simple relational classifier. In *Proceedings of MRDM-2003 at KDD-2003*, pages 64–76.

[10] S. A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *J. Mach. Learn. Res.*, 8:935–983, 2007.

[11] J. Neville and D. Jensen. Relational dependency networks. *J. Mach. Learn. Res.*, 8:653–692, 2007.

[12] J. Neville, D. Jensen, L. Friedland, and M. Hay. Learning relational probability trees. In *KDD '03*, pages 625–630, New York, NY, USA, 2003.

[13] J. Neville, D. Jensen, and B. Gallagher. Simple estimators for relational bayesian classifiers. In *ICDM '03*, page 609, 2003.

[14] P. Sen and L. Getoor. Link-based classification. Technical Report CS-TR-4858, University of Maryland, February 2007.

[15] Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD '09*, pages 797–806, 2009.

[16] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *UAI*, pages 485–492, 2002.

[17] B. Taskar, E. Segal, and D. Koller. Probabilistic classification and clustering in relational data. In *IJCAI'01*, pages 870–876, 2001.

[18] Z. Yin, R. Li, Q. Mei, and J. Han. Exploring social tagging graph for web object classification. In *KDD '09*, pages 957–966, New York, NY, USA, 2009.

[19] T. Zhang, A. Popescul, and B. Dom. Linear prediction models with graph regularization for web-page categorization. In *KDD '06*, pages 821–826, 2006.

[20] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS 16*, Vancouver, Canada, 2003.

[21] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML '03*, pages 912–919, 2003.