

# Graph Regularized Transductive Classification on Heterogeneous Information Networks<sup>\*</sup>

Ming Ji, Yizhou Sun, Marina Danilevsky, Jiawei Han and Jing Gao

Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL  
{mingji1, sun22, danilev1, hanj, jinggao3}@illinois.edu

**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 some 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 for 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 some 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, GNetMine, is proposed to model the link structure in information networks with arbitrary network schema and arbitrary 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.

## 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

---

<sup>\*</sup> Research was sponsored in part by the U.S. National Science Foundation under grant IIS-09-05215, and by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS-CTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

[11] [15] [16] [19]. Sometimes, label information is available for some data objects. Learning from labeled and unlabeled data is often called semi-supervised learning [22] [21] [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 [11] [15]. Generally, classification can be categorized into two groups: (1) transductive classification [10] [11] [22] [21] [19]: to predict labels for the given unlabeled data; and (2) inductive classification [9] [15] [12] [17] [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 networked data [9] [10] [11] [15] 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, bibliographic data naturally forms a network among papers, authors, conferences, terms, etc. 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 shopping category.

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

1. *Complexity of the network structure.* When dealing with the multi-typed network structure in a heterogeneous information network, one common solution is to transform it into a homogenous network and apply traditional classification methods [11] [15]. 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 discrim-

inative information is likely to be lost (e.g., *authors* of the paper, and the *venue* the paper is published in.) Another solution to make use of the whole network is to ignore 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 [8] [16] 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. If we transform the link information into features, we will likely generate very high dimensional and sparse data as the number of objects increases. Moreover, even if we have feature representation for some objects in a heterogeneous information network, the features of different types of objects are in different 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 apply 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 allow label propagation 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 GNetMine) is introduced in Section 4. Section 5 provides the experimental results. Finally, we conclude this work in Section 6.

## 2 Related Work

We summarize various transductive classification methods in Table 1, where one dimension represents whether the data has features/attributes or not, and

	Non-networked data	Homogenous networked data	Heterogeneous networked data
Attributed data	SVM, Logistic Regression, etc.	Statistical Relational Learning (Relational Dependency Networks, etc.)	
Non-attributed data	/	Network-only Link-based classifier, Relational Neighbor, etc.	<i>GNetMine</i>

**Table 1.** Summary of related work about transductive classification

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. 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 link information can be combined by using Naïve Bayes [4], logistic regression [9], graph regularization [20], etc. All of these methods assume that the network is homogeneous. Relational dependency networks [12] 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. [10] 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] [18], Relational Probability Trees [13] and Relational Bayesian Classifiers [14]. Macskassy et al. [11] further emphasize that homogeneity is very important for their methods to perform within-network classification well.

Recently, there has been a surge of interest in mining heterogeneous information networks [7] [2] [8] [1]. NetClus [16] 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. [19] explore social tagging graphs for heterogeneous web object classification. They construct a bipartite graph between tags and web objects to boost classification performance. Nevertheless, they fail to distinguish between different types of links. And their method is confined to the specific network schema between tags and web data, thus cannot be applied to an arbitrary link structure.

Meanwhile, graph-based learning has enjoyed long-lasting popularity in transductive classification. Most of the methods construct an affinity graph over both labeled and unlabeled examples based on local features to encode the similarity between instances. They then design a learner which preserves the smoothness and consistency over the geometrical structure of the data. Zhu et al. [22] formulate the problem using a Gaussian random field model defined with respect to

the graph. Zhou et al. [21] propose to let each point iteratively spread its label information to neighbors so as to ensure both local and global consistency. 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 extend the graph-based learning framework to fit the special characteristics of heterogeneous networked data.

### 3 Problem Definition

In this section, we introduce several related concepts and notations, and then formally define the problem.

**Definition 1. Heterogeneous information network.** Given  $m$  types of data objects, denoted by  $\mathcal{X}_1 = \{x_{11}, \dots, x_{1n_1}\}, \dots, \mathcal{X}_m = \{x_{m1}, \dots, x_{mn_m}\}$ , 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 [16]. 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  $V - V'$ .

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  $\mathcal{X}_i$ ,  $i \in \{1, \dots, m\}$ , we try to compute a class indicator matrix  $\mathbf{F}_i = [\mathbf{f}_i^{(1)}, \dots, \mathbf{f}_i^{(K)}] \in \mathbb{R}^{n_i \times K}$ , where each  $\mathbf{f}_i^{(k)} = [f_{i1}^{(k)}, \dots, f_{in_i}^{(k)}]^T$  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  $\mathbf{F}_i$ :  $c_{ip} = \arg \max_{1 \leq k \leq K} f_{ip}^{(k)}$ .

In a heterogeneous information network, a relation graph  $\mathcal{G}_{ij}$  can be built corresponding to each type of link relationships between two types of data objects  $\mathcal{X}_i$  and  $\mathcal{X}_j$ ,  $i, j \in \{1, \dots, m\}$ . 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

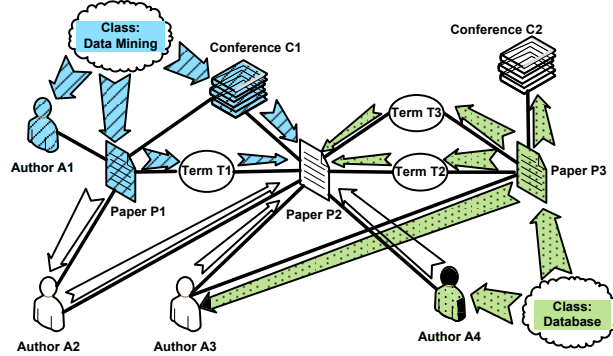


Fig. 1. Knowledge propagation in a bibliographic information network

also incorporate domain knowledge. A simple definition is as follows:

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

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  $\mathbf{y}_i^{(k)} = [y_{i1}^{(k)}, \dots, y_{in_i}^{(k)}]^T \in \mathbb{R}^{n_i}$  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  $k \in \{1, \dots, K\}$ , our goal is to infer a set of  $\mathbf{f}_i^{(k)}$  from  $\mathbf{R}_{ij}$  and  $\mathbf{y}_i^{(k)}$ ,  $i, j \in \{1, \dots, m\}$ .

## 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 database area,

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 essentially consistency preserving over the network, which is similar to [10] and [21], the interconnected relationships 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 considered 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 type differences between links and objects, 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  $f_i^{(k)}$  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,  $f_i^{(k)}$  should be as consistent as possible with the link information and prior knowledge within each relation graph, so we try to minimize the following objective function:

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

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

and its effectiveness is well proved [21]. 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  $\lambda_{ij}$  and  $\alpha_i$ , where  $0 \leq \lambda_{ij} < 1$ ,  $0 < \alpha_i < 1$ . For  $\forall i, j \in \{1, \dots, m\}$ ,  $\lambda_{ij} > 0$  indicates that object types  $\mathcal{X}_i$  and  $\mathcal{X}_j$  are linked together and this relationship is taken into consideration. The larger  $\lambda_{ij}$ , 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  $\lambda_{ij}$  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  $\alpha_i$ , to some extent, measures how much the user trusts the labels of object type  $\mathcal{X}_i$ . Similar strategy has been adopted in [8] to control the weights between different types of relations and objects. However, 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  $\mathbf{R}_{ij}$ :

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

With simple algebraic formulations, the first term of (1) can be rewritten as:

$$\begin{aligned} & \sum_{i,j=1}^m \lambda_{ij} \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} R_{ij,pq} \left( \frac{1}{\sqrt{D_{ij,pp}}} f_{ip}^{(k)} - \frac{1}{\sqrt{D_{ji,qq}}} f_{jq}^{(k)} \right)^2 \\ &= \sum_{i,j=1}^m \lambda_{ij} \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} R_{ij,pq} \left( \frac{(f_{ip}^{(k)})^2}{D_{ij,pp}} - 2 \frac{f_{ip}^{(k)} f_{jq}^{(k)}}{\sqrt{D_{ij,pp} D_{ji,qq}}} + \frac{(f_{jq}^{(k)})^2}{D_{ji,qq}} \right) \\ &= \sum_{i,j=1}^m \lambda_{ij} \left( \sum_{p=1}^{n_i} (f_{ip}^{(k)})^2 + \sum_{q=1}^{n_j} (f_{jq}^{(k)})^2 - 2 \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} (f_{ip}^{(k)} S_{ij,pq} f_{jq}^{(k)}) \right) \\ &= \sum_{i,j=1}^m \lambda_{ij} \left( (\mathbf{f}_i^{(k)})^T \mathbf{f}_i^{(k)} + (\mathbf{f}_j^{(k)})^T \mathbf{f}_j^{(k)} - 2 (\mathbf{f}_i^{(k)})^T \mathbf{S}_{ij} \mathbf{f}_j^{(k)} \right) \end{aligned} \quad (3)$$

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

$$\begin{aligned} J(\mathbf{f}_1^{(k)}, \dots, \mathbf{f}_m^{(k)}) &= \sum_{i,j=1}^m \lambda_{ij} \left( (\mathbf{f}_i^{(k)})^T \mathbf{f}_i^{(k)} + (\mathbf{f}_j^{(k)})^T \mathbf{f}_j^{(k)} - 2 (\mathbf{f}_i^{(k)})^T \mathbf{S}_{ij} \mathbf{f}_j^{(k)} \right) \\ &\quad + \sum_{i=1}^m \alpha_i (\mathbf{f}_i^{(k)} - \mathbf{y}_i^{(k)})^T (\mathbf{f}_i^{(k)} - \mathbf{y}_i^{(k)}) \end{aligned} \quad (4)$$

**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 [21]. Then we show the connection and difference between our algorithm and [21] 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 (4) reduces to:*

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

The proof can be done by simply setting  $m = 1$  in function (4). It is easy to see that the homogeneous version of our algorithm is equivalent to the objective function of [21].

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

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

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 the same as that in  $\mathbf{f}^{(k)}$ . 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 (4) is equivalent to the following:*

$$J(\mathbf{f}_1^{(k)}, \dots, \mathbf{f}_m^{(k)}) = (\mathbf{f}^{(k)})^T \mathbf{H} \mathbf{f}^{(k)} + (\mathbf{f}^{(k)} - \mathbf{y}^{(k)})^T \boldsymbol{\alpha} (\mathbf{f}^{(k)} - \mathbf{y}^{(k)}) \quad (5)$$

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

The proof can be done by considering each term in the objective function (4) separately for  $i \neq j$  and  $i = j$ , respectively, and then summing them up. Lemma 2 shows that our proposed GNetMine algorithm has a consistent form with the graph-based learning framework on homogeneous data [21], 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  $\lambda_{ij}$  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  $\lambda_{ij}$  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 at least one type of objects linking to other objects via multiple types of relationships.<sup>1</sup>

<sup>1</sup> 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  $\lambda_{12} \mathbf{L}$ .

**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 (4), which is easy to derive from equation (5):

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

$\mathbf{H}$  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  $H(J(\mathbf{f}_1^{(k)}, \dots, \mathbf{f}_m^{(k)}))$  is positive definite. Therefore, the objective function (4) is strictly convex. The unique global minimum is obtained by differentiating (4) with respect to each  $(\mathbf{f}_i^{(k)})^T$ :

$$\frac{\partial J}{\partial (\mathbf{f}_i^{(k)})^T} = \sum_{j=1, j \neq i}^m \lambda_{ij} (2\mathbf{f}_i^{(k)} - 2\mathbf{S}_{ij}\mathbf{f}_j^{(k)}) + 4\lambda_{ii}\mathbf{L}_{ii}\mathbf{f}_i^{(k)} + 2\alpha_i(\mathbf{f}_i^{(k)} - \mathbf{y}_i^{(k)}) \quad (6)$$

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

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

$$\mathbf{f}_i^{(k)} = \left( \left( \sum_{j=1, j \neq i}^m \lambda_{ij} + \alpha_i \right) \mathbf{I}_i + 2\lambda_{ii}\mathbf{L}_{ii} \right)^{-1} \left( \alpha_i \mathbf{y}_i^{(k)} + \sum_{j=1, j \neq i}^m \lambda_{ij} \mathbf{S}_{ij} \mathbf{f}_j^{(k)} \right), \quad i \in \{1, \dots, m\}$$

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

**Iterative solution** Though the closed form solution is obtained, sometimes the iterative solution is preferable. Based on equation (6), we derive the iterative form of our algorithm as follows:

- Step 0: For  $\forall k \in \{1, \dots, K\}$ ,  $\forall i \in \{1, \dots, m\}$ , initialize confidence estimates  $\mathbf{f}_i^{(k)}(0) = \mathbf{y}_i^{(k)}$  and  $t = 0$ .

- Step 1: Based on the current  $\mathbf{f}_i^{(k)}(t)$ , compute:

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

for  $\forall k \in \{1, \dots, K\}, \forall i \in \{1, \dots, m\}$ .

- Step 2: Repeat step 1 with  $t = t + 1$  until convergence, i.e., until  $\mathbf{f}_i^{(k)*} = \mathbf{f}_i^{(k)}(t)$  do not change much for all  $i$ .
- Step 3: For each  $i \in \{1, \dots, m\}$ , 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  $\mathbf{f}_i^{(k)*} = [f_{i1}^{(k)*}, \dots, f_{in_i}^{(k)*}]^T$ .

Following an analysis similar to [21], the iterative algorithm can be proven to converge to the closed form solution. The iterative solution can be viewed as a natural extension of [21], 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 GNet-Mine) on the real heterogeneous information network of DBLP<sup>2</sup>. 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.

<sup>2</sup> <http://www.informatik.uni-trier.de/~ley/db/>

## 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. By selecting five representative conferences in each area, papers published in these conferences, 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 data set we used contains 14376 papers, 20 conferences, 14475 authors and 8920 terms, with a total number of 170794 links<sup>3</sup>. By using our GNetMine algorithm, we can simultaneously classify all types of objects regardless of how many types of objects we labeled.

For accuracy evaluation, we use a labeled data set of 4057 authors, 100 papers and all 20 conferences. For more details about the labeled data set, please refer to [7] [16]. 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.

## 5.2 Algorithms for comparison

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

- Learning with Local and Global Consistency (LLGC) [21]
- Weighted-vote Relational Neighbor classifier (wvRN) [10] [11]
- Network-only Link-based classification (nLB) [9] [11]

LLGC is a graph-based transductive classification algorithm, which is also the homogenous reduction of GNetMine 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 [11], 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 heterogeneous 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

<sup>3</sup> The data set is available at [www.cs.illinois.edu/homes/mingji1/DBLP\\_four\\_area.zip](http://www.cs.illinois.edu/homes/mingji1/DBLP_four_area.zip) for sharing and experiment repeatability.

in the accuracy study. The open-source implementation of NetKit-SRL<sup>4</sup> [11] is employed in our experiments.

### 5.3 Accuracy study

In this experiment, we choose labels on both *authors* and *papers* to test the classification accuracy. In order to address the label scarcity problem, we randomly choose  $(a\%, p\%) = [(0.1\%, 0.1\%), (0.2\%, 0.2\%), \dots, (0.5\%, 0.5\%)]$  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 not even be able to extract a fully labeled sub-network for training, making many state-of-the-art algorithms inapplicable.

Since the homogeneous LLGC algorithm just has one  $\alpha$  and one  $\lambda$ , only the ratio  $\frac{\alpha}{\lambda}$  matters in the model selection. The  $\frac{\alpha}{\lambda}$  is 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 GNetMine, we do not treat any object/link type as particularly important here and use the same set of parameters as LLGC, i.e.,  $\alpha_i = 0.1$ ,  $\lambda_{ij} = 0.2$ ,  $\forall i, j \in \{1, \dots, m\}$ . This may not be the best choice, but it is good enough to show the effectiveness of GNetMine. 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 2, 3 and 4, 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 working with homogeneous data for 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 total number of objects rises to  $14376$  (*papers*) +  $20$  (*conferences*) +  $14475$  (*authors*) +  $8920$  (*terms*) =  $37791$ , out of which at most  $(14376$  (*papers*)+ $14475$  (*authors*)) $\times$   $0.5\%/37791 = 0.4\%$  objects are labeled. Similar results have been reported [11] 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

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

$(a\%, p\%)$ 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)	GNetMine (A-C-P-T)
(0.1%, 0.1%)	25.4	26.0	40.8	34.1	41.4	61.3	<b>82.9</b>
(0.2%, 0.2%)	28.3	26.0	46.0	41.2	44.7	62.2	<b>83.4</b>
(0.3%, 0.3%)	28.4	27.4	48.6	42.5	48.8	65.7	<b>86.7</b>
(0.4%, 0.4%)	30.7	26.7	46.3	45.6	48.7	66.0	<b>87.2</b>
(0.5%, 0.5%)	29.8	27.3	49.0	51.4	50.6	68.9	<b>87.5</b>

**Table 2.** Comparison of classification accuracy on authors (%)

$(a\%, p\%)$ 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)	GNetMine (A-C-P-T)
(0.1%, 0.1%)	49.8	31.5	62.0	42.0	67.2	62.7	<b>79.2</b>
(0.2%, 0.2%)	73.1	40.3	71.7	49.7	72.8	65.5	<b>83.5</b>
(0.3%, 0.3%)	77.9	35.4	77.9	54.3	76.8	66.6	<b>83.2</b>
(0.4%, 0.4%)	79.1	38.6	78.1	54.4	77.9	70.5	<b>83.7</b>
(0.5%, 0.5%)	80.7	39.3	77.9	53.5	79.0	73.5	<b>84.1</b>

**Table 3.** Comparison of classification accuracy on papers (%)

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

**Table 4.** Comparison of classification accuracy on conferences (%)

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, GNetMine performs the best on all types of objects via learning from labeled authors and papers. Even though the parameters for all types of objects and links are set to the same values, GNetMine 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.

#### 5.4 Model selection

The  $\alpha_i$ 's and  $\lambda_{ij}$ 's are essential parameters in GNetMine which control the relative importance of different terms. We empirically set all the  $\alpha_i$ 's as 0.1, and all the  $\lambda_{ij}$ 's as 0.2 in the previous experiment. In this subsection, we try to study the impact of parameters on the performance of GNetMine. Since labels are given on authors and papers, the  $\alpha_i$  associated with authors (denoted by  $\alpha_a$ ) and papers (denoted by  $\alpha_p$ ), as well as the  $\lambda_{ij}$  associated with the *author-paper* relationship (denoted by  $\lambda_{pa}$ ) are empirically more important than other parameters. So we fix all the other parameters and let  $\alpha_a$ ,  $\alpha_p$  and  $\lambda_{pa}$  vary. We also change  $\alpha$  and  $\lambda$  in LLGC accordingly. Figure 2 shows the average classification

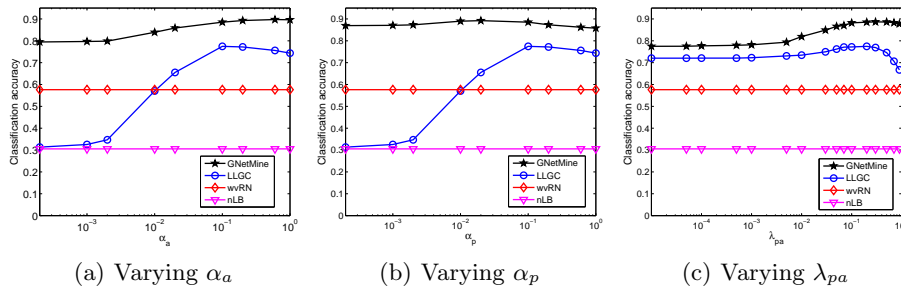


Fig. 2. Model Selection when (0.5%, 0.5%) of authors and papers are labeled

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, GNetMine 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  $\alpha_a$  is different from that of  $\alpha_p$ , indicating that authors and papers do play different roles in the classification process. Generally, the performance of GNetMine with varying  $\alpha_p$  is more stable than that with varying  $\alpha_a$ . From the accuracy curve of  $\lambda_{pa}$ , it can be seen that setting  $\lambda_{pa}$  larger than all other  $\lambda_{ij}$ 's (which are set to 0.2) improves the accuracy. This is because that increasing  $\lambda_{pa}$  enhances the knowledge propagation between the two types of labeled data, which is beneficial.

Overall, the parameter selection will not critically affect the performance of GNetMine. And if the user has some knowledge about the importance of certain types of links, the parameters can be adjusted accordingly to model the special characteristics of the network.

## 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.

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. J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han. Graph-based consensus maximization among multiple supervised and unsupervised models. In *Advances in Neural Information Processing Systems (NIPS)*, 22, pages 585–593, 2009.
8. B. Long, Z. M. Zhang, X. Wu, and P. S. Yu. Spectral clustering for multi-type relational data. In *ICML '06*, pages 585–592, 2006.
9. Q. Lu and L. Getoor. Link-based classification. In *ICML '03*, 2003.
10. S. A. Macskassy and F. Provost. A simple relational classifier. In *Proc. of MRDM-2003 at KDD-2003*, pages 64–76.
11. 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.
12. J. Neville and D. Jensen. Relational dependency networks. *J. Mach. Learn. Res.*, 8:653–692, 2007.
13. J. Neville, D. Jensen, L. Friedland, and M. Hay. Learning relational probability trees. In *KDD '03*, pages 625–630, 2003.
14. J. Neville, D. Jensen, and B. Gallagher. Simple estimators for relational bayesian classifiers. In *ICDM '03*, page 609, 2003.
15. P. Sen and L. Getoor. Link-based classification. Technical Report CS-TR-4858, University of Maryland, February 2007.
16. 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.
17. B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *UAI*, pages 485–492, 2002.
18. B. Taskar, E. Segal, and D. Koller. Probabilistic classification and clustering in relational data. In *IJCAI'01*, pages 870–876, 2001.
19. Z. Yin, R. Li, Q. Mei, and J. Han. Exploring social tagging graph for web object classification. In *KDD '09*, pages 957–966, 2009.
20. T. Zhang, A. Popescul, and B. Dom. Linear prediction models with graph regularization for web-page categorization. In *KDD '06*, pages 821–826, 2006.
21. D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS 16*, 2003.
22. X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML '03*, pages 912–919, 2003.