PathSelClus: Integrating Meta-Path Selection with User-Guided Object Clustering in Heterogeneous Information Networks

YIZHOU SUN, BRANDON NORICK, and JIAWEI HAN, University of Illinois at Urbana-Champaign
XIFENG YAN, University of California at Santa Barbara
PHILIP S. YU, University of Illinois at Chicago and King Abdulaziz University
XIAO YU, University of Illinois at Urbana-Champaign

Real-world, multiple-typed objects are often interconnected, forming heterogeneous information networks. A major challenge for link-based clustering in such networks is their potential to generate many different results, carrying rather diverse semantic meanings. In order to generate desired clustering, we propose to use *meta-path*, a path that connects object types via a sequence of relations, to control clustering with distinct semantics. Nevertheless, it is easier for a user to provide a few examples (seeds) than a weighted combination of sophisticated meta-paths to specify her clustering preference. Thus, we propose to integrate *meta-path selection* with *user-guided clustering* to cluster objects in networks, where a user first provides a small set of object seeds for each cluster as guidance. Then the system learns the weight for each meta-path that is consistent with the clustering result implied by the guidance, and generates clusters under the learned weights of meta-paths. A probabilistic approach is proposed to solve the problem, and an effective and efficient iterative algorithm, *PathSelClus*, is proposed to learn the model, where the clustering quality and the meta-path weights mutually enhance each other. Our experiments with several clustering tasks in two real networks and one synthetic network demonstrate the power of the algorithm in comparison with the baselines.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms: Algorithms

Additional Key Words and Phrases: Heterogeneous information networks, meta-path selection, user-guided clustering

ACM Reference Format:
Sun, Y., Norick, B., Han, J., Yan, X., Yu, P. S., and Yu, X. 2013. *PathSelClus*: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. *ACM Trans. Knowl. Discov. Data* 7, 3, Article 11 (September 2013), 23 pages.
DOI:http://dx.doi.org/10.1145/2500492

A preliminary version of this article appeared in the KDD conference [Sun et al. 2012].
Y. Sun is currently affiliated with Northeastern University.
The work was supported in part by the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NS-CTA), NSF IIS-09-05215, MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC, and the U.S. Air Force Office of Scientific Research MURI award FA9550-08-1-0265. The views and conclusions contained in this article are those of the authors and should not be interpreted as representing any funding agencies.
Authors' addresses: Y. Sun, College of Computer and Information Science, Northeastern University; email: yzsun@ccs.neu.edu; B. Norick, J. Han, and X. Yu, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL; X. Yan, Department of Computer Science, University of California at Santa Barbara, Santa Barbara, CA; P. S. Yu, Department of Computer Science, University of Illinois at Chicago, Chicago, IL, and Computer Science Department, King Abdulaziz University, Jeddah, Saudi Arabia.

DOI:http://dx.doi.org/10.1145/2500492

1. INTRODUCTION

With the advent of massive social and information networks, link-based clustering of objects in networks becomes increasingly important since it may help discover hidden knowledge in large networks. Link-based clustering groups objects based on their links instead of attribute values. This is especially useful when attributes of objects cannot be fully obtained. Most existing link-based clustering algorithms are on homogeneous networks where links carry the same semantic meaning and only differ in their strengths (weights). However, most real-world networks are heterogeneous, where objects are of multiple types and are linked via different types of relations or sequences of relations, forming a set of *meta-paths* [Sun et al. 2011]. These meta-paths imply diverse semantics, and thus clustering on different meta-paths will generate rather different results, as shown in the following.

Example 1.1 (*Meta-path-based clustering*). A toy heterogeneous information network is shown in Figure 1, which contains three types of objects: organization (O), author (A), and venue (V), and two types of links: the solid line represents the affiliation relation between author and organization and the dashed line, the publication relation between author and venue. Authors are then connected (indirectly) via different meta-paths. For example, $A - O - A$ is a meta-path denoting a relation between authors via organizations (colleagues), whereas $A - V - A$ denotes a relation between authors via venues (publishing in the same venues). A question then arises: which types of connections should we use to cluster the authors?

Obviously, there is no unique answer to this question. Different meta-paths lead to different author connection graphs, which may lead to different clustering results. In Figure 2(a), authors are connected via organizations and form two clusters: $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$; in Figure 2(b), authors are connected via venues and form two different clusters: $\{1, 3, 5, 7\}$ and $\{2, 4, 6, 8\}$; whereas in Figure 2(c), a connection graph combining both meta-paths generate 4 clusters: $\{1, 3\}, \{2, 4\}, \{5, 7\}$ and $\{6, 8\}$.

This toy example shows that all three clusterings look reasonable but they carry diverse semantics. It should be a user's responsibility to choose her desired meta-path(s). However, it is often difficult to ask her to explicitly specify one or a weighted combination of multiple meta-paths. Instead, it is easier for her to give some guidance in other forms, such as giving one or a couple of examples for each cluster. For example, it may not be hard to give a few known conferences in each cluster (field) if one wants to cluster them into K research areas (for a user-desired K), or ask a user to name a few restaurants if one wants to cluster them into different categories in a business review Web site (e.g., Yelp).

Methods for user-guided clustering or semi-supervised learning have been proposed on (homogeneous) graphs [Kulis et al. 2005; Zhu and Ghahramani 2002; Zhu et al. 2003]. However, since we are dealing with heterogeneous networks, these methods cannot apply. We need to explore meta-paths that represent heterogeneous connections across objects, leading to rich semantic meanings, hence diverse clustering results. With user guidance, a system will be able to learn the most appropriate meta-paths, or their weighted combinations. The learned meta-paths will in turn provide an insightful view to help understand the underlying mechanism for the formation of a specific type of clustering. For example, which meta-path is more important to determine a restaurant's category—the meta-path connecting them via customers, the one connecting them via text in reviews, or the kNN relation determined by their locations?

In this article, we integrate meta-path selection with user-guided clustering for better clustering of a user-specified type of objects, i.e., the *target objects*, in a heterogeneous information network, where the user guidance is given as a small set of seeds

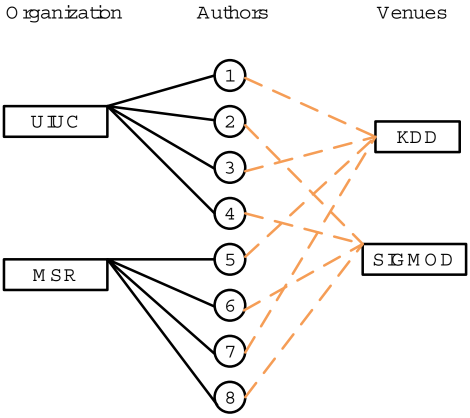
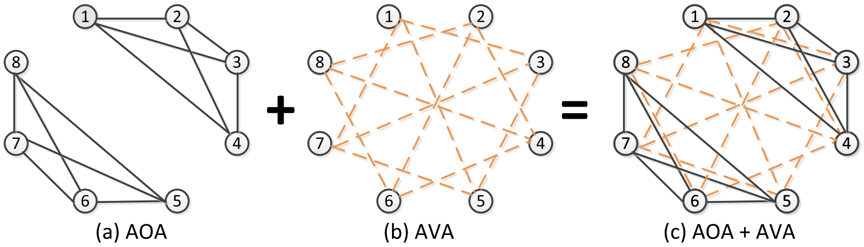


Fig. 1. A toy heterogeneous information network containing organizations, authors, and venues.



(a) AOA (b) AVA (c) AOA + AVA

Fig. 2. Author connection graphs under different meta-paths.

in each cluster. For example, to cluster authors into 2 clusters in Example 1.1, a user may seed {1} and {5} for two clusters, which implies a selection of meta-path $A - O - A$; or seed {1}, {2}, {5}, and {6} for four clusters, which implies a combination of both meta-paths $A - O - A$ and $A - V - A$ with about equal weights. Our goal is to, (1) determine the weight of each meta-path for a particular clustering task, which should be consistent with the clustering results implied by the limited user guidance, and (2) output the clustering result according to the user guidance and under the learned weight for each meta-path.

We propose a probabilistic model that models the hidden clusters for target objects, the user guidance, and the quality weights for different meta-paths in a unified framework. An effective and efficient iterative algorithm *PathSelClus* is developed to learn the model, where the clustering quality and the meta-path quality mutually enhance each other. The experiments with different tasks on two real networks and one synthetic network show our algorithm outperforms the baselines. Our contributions are summarized as follows.

(1) We propose to integrate meta-path selection with user-guided clustering for arbitrary heterogeneous networks, and study a specific form of guidance: seeding some objects in each cluster.
(2) A probabilistic model is proposed to put hidden clusters, user guidance, and the quality of meta-paths into one unified framework, and an iterative algorithm is developed where the clustering result and weights for different meta-paths are learned alternatively and mutually enhance each other.

(3) Experiments on real and synthetic heterogeneous information networks have shown the effectiveness and efficiency of our algorithm over baselines, and the learned weights of meta-paths provide knowledge for better understanding of the cluster formation.

2. PRELIMINARIES

In this section, we introduce preliminary concepts in heterogeneous information networks and define the problem of integrating meta-path selection with user-guided object clustering.

2.1. Heterogeneous Information Network

A heterogeneous information network [Sun et al. 2009a] is an information network with multiple types of objects and/or multiple types of links, formally defined in the following.

Definition 2.1. (*Information network*). An *information network* is defined as a directed graph $G = (\mathcal{V}, \mathcal{E})$ with an object type mapping function $\tau : \mathcal{V} \to \mathcal{A}$ and a link type mapping function $\phi : \mathcal{E} \to \mathcal{R}$, where each object $v \in \mathcal{V}$ belongs to one particular object type $\tau(v) \in \mathcal{A}$, each link $e \in \mathcal{E}$ belongs to a particular relation $\phi(e) \in \mathcal{R}$, and if two links belong to the same relation type, they share the same starting and ending object types.

Different from the traditional network definition, we explicitly distinguish object types and relationship types in the network. Note that, if a relation exists from type A to type B, denoted as ARB, the inverse relation R^{-1} holds naturally for $BR^{-1}A$. R and its inverse R^{-1} are usually not equal, unless the two types are the same and R is symmetric. When the types of objects $|\mathcal{A}| > 1$ or the types of relations $|\mathcal{R}| > 1$, the network is called a *heterogeneous information network*; otherwise, it is a *homogeneous information network*.

Given a complex heterogeneous information network, it is necessary to provide its meta level (schema-level) description for better understanding of the object types and link types in the network. Therefore, Sun et al. [2011] propose the concept of *network schema* to describe the meta structure of a network. The network schema of a heterogeneous information network has specified type constraints on the sets of objects and relationships between the objects. These constraints make a heterogeneous information network semistructured, guiding the exploration of the semantics of the network.

Here we introduce two heterogeneous information networks that are used in the experiment section of this article: the DBLP network and the Yelp network.

Example 2.2. (*The DBLP bibliographic network*[1]). DBLP is a typical heterogeneous information network (see schema in Figure 3(a)), which contains 4 types of objects, namely paper(P), author (A), term (T), and venue (V), including conferences and journals. Links exist between authors and papers by the relations of "write" and "written by," between papers and terms by "mention" and "mentioned by," and between venues and papers by "publish" and "published by." The "citation" relation between papers can be added further, using other data sources, such as Google scholar.

Example 2.3. (*The Yelp network*[2]). Yelp is a Web site where users can write reviews of businesses. The Yelp network (see schema in Figure 3(b)) used in this article contains 4 types of objects, namely business (B), user (U), term (T), and review (R). Links

[1] http://www.informatik.uni-trier.de/~ley/db/

[2] http://www.yelp.com/

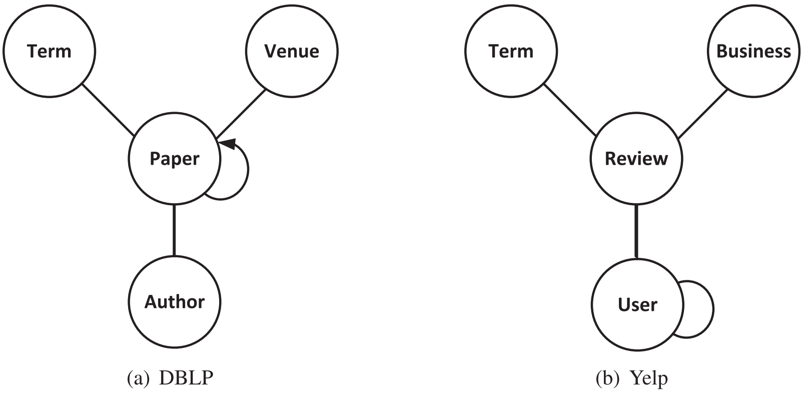


Fig. 3. Examples of heterogeneous information networks.

exist between users and reviews by the relations of "write" and "written by," between reviews and terms by "mention" and " mentioned by," between businesses and reviews by "commented by" and "comment," and between users by "friendship" (not included in our dataset).

Following Sun et al. [2011], we use the concept of *meta-path* to describe the possible relations that can be derived from a heterogeneous network between two types of objects in a meta level. Meta-path is defined by a sequence of relations in the network schema, and can be described by a sequence of object types when there is no ambiguity. For example, $A-P-A$ is a meta-path denoting the coauthorship between authors, and $A-P-V$ is a meta-path denoting the publication relation between the author and the venue type. Note that, a single relation defined in the network schema can be viewed as a special case of a meta-path, e.g., the citation relation $P \to P$.

2.2. The Meta-Path Selection Problem

Link-based clustering is to cluster objects based on their connections to other objects in the network. In a heterogeneous information network, we need to specify more information for a meaningful clustering.

First, we need to specify the type of objects we want to cluster, which is called the *target type*. Second, we need to specify which type of connection, i.e., meta-path, to use for the clustering task, where we call the object type that the target type is connecting to via the meta-path as the *feature type*. For example, when clustering authors based on the venues they have published papers in, the target type is the author type, the meta-path to use is $A-P-V$, and the feature type is the venue type; when clustering venues based on venues that share common authors, the target type is the venue type, the meta-path to use is $V-P-A-P-V$, and the feature type is still the venue type.

In a heterogeneous information network, target objects could link to many types of feature objects by multiple meta-paths. For example, authors could connect to other authors by meta-path $A-P-A$, or connect to terms by meta-path $A-P-T$. The meta-path selection problem is then to determine which meta-paths or their weighted combination to use for a specific clustering task.

2.3. User-Guided Clustering

User guidance is critical for clustering objects in the network. As shown in the motivating example, by using different types of link information in a network, different reasonable clustering results can be generated. It is users' responsibility to specify which clustering result is their desired one.

In this study, we consider the guidance in the form of object seeds in each cluster given by users. For example, to cluster authors based on their hidden research areas, one can first provide several representative authors as seeds in each area. On one hand, these seeds are used as guidance for clustering all the target objects in the network. On the other hand, they provide information for selecting the most relevant meta-paths for the specific clustering task. Note that in practice, a user may not be able to provide seeds for every cluster, but only for some clusters they are most familiar with. This situation should be handled by the algorithm too.

2.4. The Problem Definition

In all, given a heterogeneous information network G, a user needs to specify the following as inputs for a clustering task.

(1) The target type for clustering, type T, must be specified;
(2) the number of clusters, K, and the object seeds for each cluster, say $\mathcal{L}_1, \ldots, \mathcal{L}_K$, where $\mathcal{L}_k$ denotes the object seeds for cluster k, which could be an empty set; these seeds will be used as the hints to learn the purpose/preference of the clustering task;
(3) a set of M meta-paths starting from type T, denoted as $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_M$, which might be helpful for the clustering task. These meta-paths can be determined either according to users' expert knowledge, or by traversing the network schema starting from type T with a length constraint.

For each meta-path $\mathcal{P}_m$, we calculate the adjacency matrix W_m, which we call *relation matrix*, between the target type T and the feature type F_m, by multiplying adjacency matrices for each relation along the meta-path. For example, the relation matrix W for meta-path $A - P - V$, denoting the number of papers published by an author in a venue, is calculated by $W = W_{AP} \times W_{PV}$, where W_{AP} and W_{PV} are the adjacency matrices for relation $A - P$ and $P - V$ respectively.

The output of the algorithm includes two parts:

(1) to determine the weight $\alpha_m \geq 0$ of each meta-path $\mathcal{P}_m$ for a particular clustering task, which should be consistent with the clustering result implied by the limited user guidance; and
(2) to output the clustering result according to the user's guidance and under the learned weights for each meta-path, that is, to associate each target object t_i in T with a K-dimensional soft clustering probability vector, $\boldsymbol{\theta}_i = (\theta_{i1}, \ldots, \theta_{iK})$, where θ_{ik} is the probability of t_i belonging to cluster k, i.e., $\theta_{ik} \geq 0$ and $\sum_{k=1}^{K} \theta_{ik} = 1$.

3. THE PROBABILISTIC MODEL

In this section, we propose a probabilistic approach to model the problem into a unified framework. We assign probabilities for different possible clustering configurations for each target object and quality weights for each meta-path, and the goal is to find the most likely clustering result and quality weights under the probabilistic model definition.

A good clustering result is determined by several factors: first, the clustering result should be consistent with the link structure, which is determined the meta-paths; second, the clustering result should be consistent with the user guidance; and third, the quality weight of each meta-path is implied by the user-guided clustering, which should be modeled and learned to further enhance the clustering quality.

In the following, we first introduce the modeling for the three aspects respectively, and then propose a unified model that takes all of them into consideration.

3.1. Modeling the Relationship Generation

To model the consistency between a clustering result and a meta-path-derived link structure, we propose a clustering-based generative model for relationship generation.

For a meta-path $\mathcal{P}_m$, let its corresponding relation matrix between the target type T and the feature type F_m be W_m. For each target object t_i, we model its relationships as generated from a mixture of multinomial distributions, where the probability of $t_i \in T$ connecting to $f_{j,m} \in F_m$ is conditionally independent on t_i given the hidden cluster label of the relationship is known. Let $\pi_{ij,m} = P(j|i,m)$ be the generative probability of the relationship starting from t_i and ending at $f_{j,m}$, where $\sum_j \pi_{ij,m} = 1$, then

$$\pi_{ij,m} = P(j|i,m) = \sum_k P(k|i)P(j|k,m) = \sum_k \theta_{ik}\beta_{kj,m}, \tag{1}$$

where $\theta_{ik} = P(k|i)$ denotes the probability of t_i belonging to cluster k and $\beta_{kj,m} = P(j|k,m)$ denotes the probability of $f_{j,m}$ appearing in cluster k. In other words, let $\boldsymbol{\pi}_{i,m} = (\pi_{i1,m}, \ldots, \pi_{i|F_m|,m})$ be the generative probability vector for target object t_i, then each $\boldsymbol{\pi}_{i,m}$ can be factorized as a weighted summation of ranking distributions of feature objects in each cluster. The factorization idea is similar to that of PLSA [Hofmann 1999], PHITS [Cohn and Chang 2000], and RankClus [Sun et al. 2009a], but is built on meta-path-encoded relationships rather than immediate links. This extension will capture more and richer link-based features for clustering target objects in heterogeneous networks.

By assuming each target object t_i is independent and each relationship generated by t_i is independent, conditional on that their clustering configuration is known, the probability of observing all the relationships between all the target objects and feature objects is the production of the probabilities of all the relationships following meta-path $\mathcal{P}_m$:

$$P(W_m|\Pi_m, \Theta, B_m) = \prod_i P(\mathbf{w}_{i,m}|\boldsymbol{\pi}_{i,m}, \Theta, B_m) = \prod_i \prod_j (\pi_{ij,m})^{w_{ij,m}}, \tag{2}$$

where $\Pi_m = \Theta B_m$ is the probability matrix with cells as $\pi_{ij,m}$'s, Θ is the parameter matrix for θ_{ik}'s, B_m is the parameter matrix for $\beta_{kj,m}$'s, and $w_{ij,m}$ is the weight of the relationship between t_i and $f_{j,m}$. Note that, each meta-path $\mathcal{P}_m$ corresponds to a different generative probability matrix Π_m to model the relationship generation. The factorization of these probability matrices share the same soft clustering probabilities Θ, but different ranking distributions B_m in different meta-paths.

How to define and determine the weight for each meta-path in the clustering process for target objects is then very critical, and will be introduced in Section 3.3.

3.2. Modeling the Guidance from Users

Further, we take user guidance in the form of object seeds for some clusters as the prior knowledge for the clustering result Θ, by modeling the prior as a Dirichlet distribution rather than treating them as hard labeled ones.

For each target object t_i, its clustering probability vector $\boldsymbol{\theta}_i$ is assumed to be a multinomial distribution, which is generated from some Dirichlet distribution. If t_i is labeled as a seed in cluster k^*, $\boldsymbol{\theta}_i$ is then modeled as being sampled from a Dirichlet distribution with parameter vector $\lambda\mathbf{e}_{k^*} + \mathbf{1}$, where $\mathbf{e}_{k^*}$ is a K-dimensional basis vector, with the k^*th element as 1 and 0 elsewhere. If t_i is not a seed, $\boldsymbol{\theta}_i$ is then assumed as being sampled from a uniform distribution, which can also be viewed as a Dirichlet distribution with parameter vector of $\mathbf{1}$. The density of $\boldsymbol{\theta}_i$ given such priors is:

$$P(\boldsymbol{\theta}_i|\lambda) \propto \begin{cases} \prod_k \theta_{ik}^{\mathbf{1}_{\{t_i \in \mathcal{L}_k\}}\lambda} = \theta_{ik^*}^{\lambda}, & \text{if } t_i \text{ is labeled and } t_i \in \mathcal{L}_{k^*}, \\ 1, & \text{if } t_i \text{ is not labeled.} \end{cases} \tag{3}$$

where $\mathbf{1}_{\{t_i \in \mathcal{L}_k\}}$ is an indicator function, which is 1 if $t_i \in \mathcal{L}_k$ holds, and 0 otherwise.

The hyperparameter λ is a nonnegative value, which controls the strength of users' confidence over the object seeds in each cluster. From Equation (3), we can find that:

- when $\lambda = 0$, the prior for $\boldsymbol{\theta}_i$ of a labeled target object becomes a uniform distribution, which means no guidance information will be used in the clustering process;
- when $\lambda \to \infty$, the prior for $\boldsymbol{\theta}_i$ of a labeled target object converges to a point mass, i.e., $P(\boldsymbol{\theta}_i = \mathbf{e}_{k^*}) \to 1$ or $\boldsymbol{\theta}_i \to \mathbf{e}_{k^*}$, which means we will assign k^* as the hard cluster label for t_i.

In general, a larger λ indicates a higher probability that $\boldsymbol{\theta}_i$ is around the point mass $\mathbf{e}_{k^*}$, and thus a higher confidence for the user guidance.

3.3. Modeling the Quality Weights for Meta-Path Selection

Different meta-paths may lead to different clustering results, therefore it is desirable to learn the quality for each meta-path for the specific clustering task. We propose to learn the quality weight for each meta-path by evaluating the consistency between its relation matrix and the user-guided clustering result.

In deciding the clustering result for target objects, a meta-path may be of low quality for the following reasons.

(1) The relation matrix derived by the meta-path does not contain an inherent cluster structure. For example, target objects connect to the feature objects randomly.
(2) The relation matrix derived by the meta-path itself has a good inherent cluster structure, however, it is not consistent with the user guidance. For example, in our motivating example, if the user gives guidance as: $K = 2, \mathcal{L}_1 = \{1\}, \mathcal{L}_2 = \{2\}$, then the meta-path $A - O - A$ should have a lower impact in the clustering process for authors.

The general idea of measuring the quality of each meta-path is to see whether the relation matrix W_m is consistent with the detected hidden clusters Θ and thus the generative probability matrix Π_m, which is a function of Θ, i.e., $\Pi_m = \Theta B_m$. The higher the consistency of W_m with Π_m, the higher posterior probability of $P(\Pi_m|W_m)$ should be.

In order to quantify the weight for such quality, we model the weight α_m for meta-path $\mathcal{P}_m$ as the *relative weight* for each relationship between target objects and feature objects following $\mathcal{P}_m$. In other words, we treat our observations of the relation matrix as $\alpha_m W_m$ rather than the original W_m. A larger α_m indicates a higher quality and a higher confidence of the observed relationships, and thus each relationship should count more.

Then, we assume the multinomial distribution $\boldsymbol{\pi}_{i,m}$ has a prior of a Dirichlet distribution with parameter vector $\phi_{i,m}$. In this article, we consider a discrete uniform prior, which is a special case of Dirichlet distribution with parameters as an all-one vector, $\phi_{i,m} = \mathbf{1}$. The value of α_m is determined by the consistency between the observed relation matrix W_m and the generative probability matrix Π_m. The goal is to find the α_m^*

that maximizes the posterior probability of $\boldsymbol{\pi}_{i,m}$ for all the target objects t_i, given the observation of relationships $\mathbf{w}_{i,m}$ with relative weight α_m:

$$\alpha_m^* = \arg\max_{\alpha_m} \prod_i P(\boldsymbol{\pi}_{i,m} | \alpha_m \mathbf{w}_{i,m}, \boldsymbol{\theta}_i, B_m). \tag{4}$$

The posterior of $\boldsymbol{\pi}_{i,m} = \boldsymbol{\theta}_i B_m$ is another Dirichlet distribution with the updated parameter vector as $\alpha_m \mathbf{w}_{i,m} + \mathbf{1}$, according to the multinomial-Dirichlet conjugate:

$$\boldsymbol{\pi}_{i,m} | \alpha_m \mathbf{w}_{i,m}, \boldsymbol{\theta}_i, B_m \sim Dir(\alpha_m w_{ij,m} + 1, \ldots, \alpha_m w_{i|F_m|,m} + 1), \tag{5}$$

which has the following density function:

$$P(\boldsymbol{\pi}_{i,m} | \alpha_m \mathbf{w}_{i,m}, \boldsymbol{\theta}_i, B_m) = \frac{\Gamma(\alpha_m n_{i,m} + |F_m|)}{\prod_j \Gamma(\alpha_m w_{ij,m} + 1)} \prod_j (\pi_{ij,m})^{\alpha_m w_{ij,m}}, \tag{6}$$

where $n_{i,m} = \sum_j w_{ij,m}$ is the total number of path instances from t_i following meta-path $\mathcal{P}_m$.

By modeling α_m in such a way, the meaning of α_m is quite clear.

— $\alpha_m w_{ij,m} + 1$ is the parameter of jth dimension for the new Dirichlet distribution.

— The larger α_m, the more likely it will generate a $\boldsymbol{\pi}_{i,m}$ with a distribution as the observed relationship distribution, i.e., $\boldsymbol{\pi}_{i,m} \rightarrow \mathbf{w_{i,m}}/n_{i,m}$ when $\alpha_m \rightarrow \infty$, where $n_{i,m}$ is the total number of path instances from t_i following meta-path $\mathcal{P}_m$.

— The smaller α_m, the more likely it will generate a $\boldsymbol{\pi}_i$ that is not relevant to the relation matrix W_m, and $\boldsymbol{\pi}_{i,m}$ can be any $|F_m|$-dimensional multinomial distribution.

Note that, we do not consider negative α_m's in this model, which means relationships with a negative impact in the clustering process are not considered, and the extreme case of $\alpha_m = 0$ means the relationships in a meta-path are totally irrelevant for the clustering process.

Discussion of the Prior of $\boldsymbol{\pi}_{i,m}$. In this article, we assume $\boldsymbol{\pi}_{i,m}$ has a Dirichlet prior with parameters as an all-one vector, that is, a discrete uniform distribution. In practice, we may vary the parameters, depending on our different assumptions on the unstructured component of the relationship generation. For example, we may assume $\boldsymbol{\pi}_{i,m}$ follows a symmetric Dirichlet prior with a high concentration parameter, indicating that we assume by default the relationships are generated totally randomly (uniformly); or we may assume it follows a Dirichlet prior with parameters proportional to the empirical distribution of the feature objects in the meta-path, indicating that we assume by default the relationships are generated with such a background distribution.

3.4. The Unified Model

By putting all three factors together, we have the joint probability of observing the relation matrices with relative weights α_m's, and the parameter matrices Π_m's and Θ.

$$\begin{aligned} &P(\{\alpha_m W_m\}_{m=1}^M, \Pi_{1:M}, \Theta | B_{1:M}, \Phi_{1:M}, \lambda) \\ &= \prod_i \Big(\prod_m P(\alpha_m \mathbf{w}_{i,m} | \boldsymbol{\pi}_{i,m}, \boldsymbol{\theta}_i, B_m) P(\boldsymbol{\pi}_{i,m} | \phi_{i,m})\Big) P(\boldsymbol{\theta}_i | \lambda), \end{aligned} \tag{7}$$

where Φ_m is the Dirichlet prior parameter matrix for Π_m, and an all-one matrix in our case. We want to find the maximum a posteriori probability (MAP) estimate for Π_m's and Θ, that maximizes the logarithm of posterior probability of $\{\Pi_m\}_{m=1}^M$, given the observations of relation matrices with relative weights $\{\alpha_m W_m\}_{m=1}^M$ and Θ, plus a regularization term over $\boldsymbol{\theta}_i$ for each target object denoting the logarithm of prior density of $\boldsymbol{\theta}_i$:

$$J = \sum_i \Big(\sum_m \log P(\boldsymbol{\pi}_{i,m}|\alpha_m \mathbf{w}_{i,m}, \boldsymbol{\theta}_i, B_m) + \sum_k \mathbf{1}_{\{t_i \in \mathcal{L}_k\}} \lambda \log \theta_{ik}\Big). \tag{8}$$

By substituting the posterior probability formula in Equation (6) and the factorization form for all $\boldsymbol{\pi}_{i,m}$, we get the final objective function:

$$\begin{aligned} J = \sum_i \Big(\sum_m \Big(\sum_j \alpha_m w_{ij,m} \log \sum_k \theta_{ik} \beta_{kj,m} \\ + \log \Gamma(\alpha_m n_{i,m} + |F_m|) - \sum_j \log \Gamma(\alpha_m w_{ij,m} + 1)\Big) \\ + \sum_k \mathbf{1}_{\{t_i \in \mathcal{L}_k\}} \lambda \log \theta_{ik} \Big). \end{aligned} \tag{9}$$

4. THE LEARNING ALGORITHM

In this section, we introduce the learning algorithm, *PathSelClus*, for the model (Equation (9)) proposed in Section 3. It is a two-step iterative algorithm, where the clustering result Θ and the weights for each meta-path $\boldsymbol{\alpha}$ mutually enhance each other. In the first step, we fix the weight vector $\boldsymbol{\alpha}$, and learn the best clustering results Θ under this weight. In the second step, we fix the clustering matrix Θ and learn the best weight vector $\boldsymbol{\alpha}$.

4.1. Optimize Θ Given α

When $\boldsymbol{\alpha}$ is fixed, the terms only involving $\boldsymbol{\alpha}$ can be discarded in the objective function Equation (9), which is then reduced to:

$$J_1 = \sum_m \alpha_m \sum_i \sum_j w_{ij,m} \log \sum_k \theta_{ik} \beta_{kj,m} + \sum_i \sum_k \mathbf{1}_{\{t_i \in \mathcal{L}_k\}} \lambda \log \theta_{ik}. \tag{10}$$

The new objective function can be viewed as a weighted summation of the log-likelihood for each relation matrix under each meta-path, where the weight α_m indicates the quality of each meta-path, plus a regularization term over Θ representing the user guidance.

Θ and the augmented parameter B_m's can be learned using the standard EM algorithm, as follows.

E-step. In each relation matrix, we use $z_{ij,m}$ to denote the cluster label for each relationship between a target object t_i and a feature object $f_{j,m}$. According to the generative process described in Section 3.1, $P(z_{ij,m} = k) = \theta_{ik}$, and $f_{j,m}$ is picked with probability $\beta_{kj,m}$. The conditional probability of the hidden cluster label given the old Θ^{t-1} and B_m^{t-1} values is:

$$p(z_{ij,m} = k|\Theta^{t-1}, B_m^{t-1}) \propto \theta_{ik}^{t-1} \beta_{kj,m}^{t-1}. \tag{11}$$

The Q-function $Q(\Theta, B_m|\Theta^{t-1}, B_m^{t-1})$, which is the tight lower bound of J_1 according to Jensen's inequality, is then:

$$\begin{aligned} &Q(\Theta, B_m|\Theta^{t-1}, B_m^{t-1}) \\ &= \sum_m \alpha_m \sum_i \sum_j w_{ij,m} \sum_k p(z_{ij,m} = k|\Theta^{t-1}, B_m^{t-1}) \log \theta_{ik}\beta_{kj,m} + \sum_i \sum_k \mathbf{1}_{\{t_i \in \mathcal{L}_k\}} \lambda \log \theta_{ik}. \end{aligned} \tag{12}$$

M-step. By maximizing the Q-function, we have the updating formulas for Θ^t and B_m^t as:

$$\theta_{ik}^t \propto \sum_m \alpha_m \sum_j w_{ij,m} p(z_{ij,m} = k|\Theta^{t-1}, B_m^{t-1}) + \mathbf{1}_{\{t_i \in \mathcal{L}_k\}} \lambda \tag{13}$$

$$\beta_{kj,m}^t \propto \sum_j w_{ij,m} p(z_{ij,m} = k|\Theta^{t-1}, B_m^{t-1}). \tag{14}$$

From Equation (13), we can see that the clustering membership vector $\boldsymbol{\theta}_i$ for t_i is determined by the cluster labels of all its relationships to feature objects, in all the relation matrices. Besides, if t_i is labeled as a seed object in some cluster k^*, $\boldsymbol{\theta}_i$ is also determined by the label. The strength of impacts from these factors is determined by the weight of each meta-path α_m, and the strength of the cluster labels λ, where α_m's are learned automatically by our algorithm, and λ is given by users.

4.2. Optimize α Given Θ

Once given a clustering result Θ and the augmented parameter B_m's, we can calculate the generative probability matrix Π_m for each meta-path $\mathcal{P}_m$ by: $\Pi_m = \Theta B_m$. By discarding the irrelevant terms, the objective function of Equation (9) can be reduced to:

$$J_2 = \sum_i \Big(\sum_m \big(\sum_j \alpha_m w_{ij,m} \log \pi_{ij,m} + \log \Gamma(\alpha_m n_{i,m} + |F_m|) - \sum_j \log \Gamma(\alpha_m w_{ij,m} + 1) \big) \Big). \tag{15}$$

It is easy to prove that J_2 is a concave function, which means there is a unique $\boldsymbol{\alpha}$ that maximizes J_2. We use the gradient descent approach to solve the problem, which is an iterative algorithm with the updating formula as:

$$\alpha_m^t = \alpha_m^{t-1} + \eta_m^t \left. \frac{\partial J_2}{\partial \alpha_m} \right|_{\alpha_m = \alpha_m^{t-1}},$$

where the partial derivative of α_m can be derived as:

$$\frac{\partial J_2}{\partial \alpha_m} = \sum_i \sum_j w_{ij,m} \log \pi_{ij,m} + \sum_i \psi(\alpha_m n_{im} + |F_m|) n_{i,m} - \sum_i \sum_j \psi(\alpha_m w_{ij,m} + 1) w_{ij,m},$$

where $\psi(x)$ is the digamma function, the first derivative of $\log \Gamma(x)$.

The step size η_m^t is usually set as a small enough number, to guarantee the increase of J_2. In this article, we follow the trick used in nonnegative matrix factorization (NMF) [Lee and Seung 2000], and set

$$\eta_m^t = \frac{\alpha_m^{t-1}}{-\sum_i \sum_j w_{ij,m} \log \pi_{ij,m}}.$$

Algorithm 1. The *PathSelClus* Algorithm.

```
Input: Network: $G$, Metapath: $\{\mathcal{P}\}_{m=1}^{M}$, Number of cluster: $K$, Object seeds: $\{\mathcal{L}_1, \ldots, \mathcal{L}_K\}$,
       User belief: $\lambda$;
Output: The clustering result $\Theta$; the weight vector for meta-paths $\boldsymbol{\alpha}$;
Normalize the weight of each relation matrix $W_m$ into $\frac{W_m}{\sum_{ij} W_{ij,m}}$;
$\boldsymbol{\alpha} = \mathbf{1}$;
repeat
    Initialize $\Theta^0$ and $B^0$;
    repeat
        1. E-step: update $p(z_{ij,m} = k | \Theta^{t-1}, B_m^{t-1})$ by Equation (11);
        2. M-step: update $\Theta^t$ and $B_m^t$ by Equations (13) and (14);
    until reaches cluster change threshold;
    $\Theta = \Theta^t$ ;
    $\boldsymbol{\alpha}^0 = \boldsymbol{\alpha}$ ;
    repeat
        1. update $\boldsymbol{\alpha}^t$ by Equation (16) ;
    until reaches inner $\boldsymbol{\alpha}$ difference threshold;
    $\boldsymbol{\alpha} = \boldsymbol{\alpha}^t$ ;
until reaches $\boldsymbol{\alpha}$ difference threshold;
Output $\Theta$ and $\boldsymbol{\alpha}$;
```

By using this step size, we can get the updating formula for α_m as:

$$\alpha_m^t = \alpha_m^{t-1} \frac{\sum_i \big(\psi(\alpha_m^{t-1} n_{im} + |F_m|) n_{i,m} - \sum_j \psi(\alpha_m^{t-1} w_{ij,m} + 1) w_{ij,m}\big)}{-\sum_i \sum_j w_{ij,m} \log \pi_{ij,m}}, \tag{16}$$

which is guaranteed to be a nonnegative value.

Also, by looking at the denominator of the formula, we can see there is a larger log-likelihood of observing relationships $w_{ij,m}$ under model probability $\pi_{ij,m}$, which means a smaller denominator as log-likelihood is negative, generally leads to a larger α_m. This is also consistent with human intuition.

4.3. The PathSelClus Algorithm

The *PathSelClus* algorithm is then summarized in Algorithm 1. Overall, it is an iterative algorithm that optimizes Θ and $\boldsymbol{\alpha}$ alternatively. The optimization of Θ contains an inner loop of an EM-algorithm, and the optimization of $\boldsymbol{\alpha}$ contains another inner loop of a gradient descent algorithm. We discuss some details of the algorithm implementation in the following.

4.3.1. The Weight Setting of Relation Matrices. Given a heterogeneous information network G, we calculate the relation matrix W_m for each given meta-path $\mathcal{P}_m$ by multiplying adjacency matrices along the meta-path. It can be shown that, scaling W_m by a factor of $1/c_m$ leads to a scaling of the learned relative weight α_m by a factor of c_m. Therefore, the performance of the clustering result will not be affected by the scaling of the relation matrix, which is a good property of our algorithm. In the experiments, we normalize each W_m by its total weight, so that the initial contributions from each meta-path are comparable to each other.

4.3.2. Initialization Issues. For the initial value of $\boldsymbol{\alpha}$, we set it as an all-one vector, which assumes all the meta-paths are equally important. For the initial value of Θ in the clustering step given $\boldsymbol{\alpha}$, if t_i is not labeled, we assign a random clustering vector to $\boldsymbol{\theta}_i$; while if t_i is labeled as a seed for a cluster k^*, we assign $\boldsymbol{\theta}_i = \mathbf{e}^*_k$.

4.3.3. Time Complexity Analysis. The *PathSelClus* algorithm is very efficient, as it is proportional to the number of relationships that are used in the clustering process, which is about linear to the number of target objects for short meta-paths in sparse networks.

Formally, for the inner EM algorithm that optimizes Θ, the time complexity is $O(t_1(K\sum_m |E_m| + K|T| + K\sum_m |F_m|)) = O(t_1(K\sum_m |E_m|))$, where $|E_m|$ is the number of nonempty relationships in relation matrix W_m, $|T|$ and $|F_m|$ are the numbers of target objects and feature objects in meta-path $\mathcal{P}_m$, which are typically smaller than $|E_m|$, and t_1 is the number of iterations. For the inner gradient descent algorithm, the time complexity is $O(t_2(\sum_m |E_m|))$, where t_2 is the number of iterations. The total time complexity for the whole algorithm is then $O(t(t_1(K\sum_m |E_m|) + t_2(\sum_m |E_m|)))$, where t is the number of outer iterations, which usually is a small number.

5. EXPERIMENTS

In this section, we compare *PathSelClus* with several baselines, and show the effectiveness and efficiency of our algorithm.

5.1. Datasets

Two real information networks, the DBLP network and the Yelp network, and one synthetic network are used for performance study. For each network, we design multiple clustering tasks provided with different user guidance, as introduced in the following.

(1) *The DBLP Network.* For the DBLP network introduced in Example 2.1, we design three clustering tasks in the following:
- —DBLP-T1. Cluster conferences in the “four-area dataset” [Sun et al. 2009b], which contains 20 major conferences and all the related papers, authors and terms in DM, DB, IR, and ML fields, according to the research areas of the conferences. The candidate meta-paths include: $V-P-A-P-V$ and $V-P-T-P-V$.
- —DBLP-T2. Cluster top-2000 authors (by their number of publications) in the “four-area dataset”, according to their research areas. The candidate meta-paths include: $A-P-A$, $A-P-A-P-A$, $A-P-V-P-A$, and $A-P-T-P-A$.
- —DBLP-T3. Cluster 165 authors who have ever been advised by Christos Faloutsos, Michael I. Jordan, Jiawei Han, and Dan Roth (including these professors), according to their research groups. The candidate meta-paths are the same as in DBLP-T2.

(2) *The Yelp Network.* For the Yelp network introduced in Example 2.2, we are provided by Yelp a subnetwork,[3] which includes 6900 businesses, 152,327 reviews, and 65,888 users. Hierarchical categories are provided for each business as well, such as “Restaurants” and “Shopping.” For the Yelp network, we design three clustering tasks in the following.
- —Yelp-T1. We select 4 relatively big categories (“Health and Medical”, “Food”, “Shopping”, and “Beauty and Spas”), and cluster 2224 businesses with more than one review according to two meta-paths: $B-R-U-R-B$ and $B-R-T-R-B$.

[3] http://www.yelp.com/academic_dataset;
http://engineeringblog.yelp.com/2011/09/calling-all-data-miners.html

- Yelp-T2. We select 6 relatively big subcategories under the first-level category "Restaurant" ("Sandwiches", "Thai", "American (New)", "Mexican", "Italian", and "Chinese"), and cluster 554 businesses with more than one review according to the same two meta-paths.
- Yelp-T3. We select 6 relatively big subcategories under the first-level category "Shopping" ("Eyewear & Opticians", "Books, Mags, Music and Video", "Sporting Goods", "Fashion", "Drugstores", and "Home & Garden"), and cluster 484 businesses with more than one review according to the same two meta-paths.

(3) *Synthetic Network.* In addition to the two real networks, we also construct synthetic networks as the test dataset, for which the ground truth labels are given. Specifically, we generate the network according to the relationship generation model, by fixing the Θ and B_m matrices for all the target objects and each meta-path.

- Synthetic-T1. In Task 1, we generate 3 relation matrices for 1000 target objects, under the same Θ but different B_m (1000, 800, 800 feature objects respectively), which means all these relation matrices have the same underlying clustering structure. However, we generate relationships by adding a uniform distraction (noise) in addition to $\Pi_m = \Theta B_m$, each relation matrix with a different level of noise (80%, 20%, 10%). This task tests whether *PathSelClus* will assign lower weights to lower quality (more noise) relation matrices, and therefore improves the clustering accuracy.
- Synthetic-T2. In Task 2, we still generate 3 relation matrices for 1000 target objects, but with different Θ_m and B_m (all with 800 feature objects). The seeds are only generated according to Θ_1, i.e., we want to cluster target objects according to the first relation matrix. This task tests whether *PathSelClus* will assign lower weights to relation matrices that are irrelevant to the user guidance, and therefore improves the clustering results that meet users' demands.

5.2. Effectiveness Study

First, we study the effectiveness of our algorithm for different tasks, and compare it with several baselines.

5.2.1. Baselines. Three baselines are used in this study. Since none of them has considered the meta-path selection problem, we will use all the meta-paths as features and prepare them to fit the input of each of these algorithms. The first one is user-guided information theoretic-based k-means clustering (ITC), which is an adaption of seeded k-means algorithm proposed in Basu et al. [2002], by replacing Euclidean distance to KL-divergence as used in information theoretic-based clustering algorithms [Banerjee et al. 2005; Dhillon et al. 2003]. ITC is a hard clustering algorithm. For the input, we concatenate all the relation matrices side-by-side into one single relation matrix, and thus we get a very high dimensional feature vector for each target object.

The second baseline is the label propagation (LP) algorithm proposed in Zhu et al. [2003], which utilizes link structure to propagate labels to the rest of the network. For the input, we add all the relation matrices together to get one single relation matrix. As LP is designed for homogeneous networks, we confine our meta-paths to ones that start and end both in the target type. LP is a soft clustering algorithm.

The third baseline is the cluster ensemble algorithm proposed in Punera and Ghosh [2008], which can combine soft clustering results into a consensus, which we call ensemble_soft. Differently from the previous two baselines that directly combine meta-paths at the input level, cluster ensemble combines the clustering results for different meta-paths at the output level. Besides, we also use majority voting as another baseline (ensemble_voting), which first maps each clustering result for each target object

Table I. Clustering Accuracy for DBLP Tasks: DBLP-T1

#S	Measure	PathSelClus	LP	ITC	LP_voting	LP_soft	ITC_voting	ITC_soft
1	Accuracy	**0.9950**	0.6500	0.6900	0.6500	0.6650	0.6450	0.5100
	NMI	**0.9906**	0.6181	0.6986	0.6181	0.5801	0.5903	0.5316
2	Accuracy	**1**	0.7500	0.8450	0.7500	0.8200	0.8950	0.8700
	NMI	**1**	0.6734	0.7752	0.6734	0.7492	0.8321	0.7942

Table II. Clustering Accuracy for DBLP Tasks: DBLP-T2

#S	Measure	PathSelClus	LP	ITC	LP_voting	LP_soft	ITC_voting	ITC_soft
1	Accuracy	**0.7951**	0.2122	0.3284	0.2109	0.3529	0.2513	0.2548
	NMI	**0.6770**	0.0312	0.1277	0.0267	0.0301	0.4317	0.4398
5	Accuracy	**0.8815**	0.2487	0.3223	0.5117	0.3685	0.3311	0.3495
	NMI	**0.6868**	0.0991	0.1102	0.4402	0.0760	0.3092	0.4316
10	Accuracy	**0.8863**	0.5586	0.3694	0.4297	0.3880	0.4891	0.2969
	NMI	**0.6947**	0.4025	0.1261	0.1788	0.1148	0.4045	0.4204

into a hard cluster label and then picks the cluster label that is the majority over different meta-paths. As we can use either ITC or LP as the clustering algorithm for each ensemble method, we then get four ensemble baselines in total: ITC_soft, ITC_voting, LP_soft, and LP_voting.

5.2.2. Evaluation Methods. Two evaluation methods are used to test the clustering result compared with the ground truth, where the soft clustering is mapped into hard cluster labels.

The first measure is *accuracy*, which is used when seeds are available for every cluster and is calculated as the percentage of target objects going to the correct cluster. Note that, in order to measure whether the seeds are indeed attracting objects to the right cluster, we do not map the outcome cluster labels to the given class labels.

The second measure is *normalized mutual information (NMI)*, which does not require the mapping relation between ground truth labels and the cluster labels obtained by the clustering algorithm. The normalized mutual information of two partitions X and Y is calculated as: $NMI(X,Y) = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}$, where X and Y are vectors containing cluster labels for all the target objects.

Both measures are in the range of 0 to 1: a higher value indicates a better clustering result in terms of the ground truth.

5.2.3. Full Cluster Seeds. We first test the clustering accuracy when cluster seeds are given for every cluster. Since LP can only work on homogeneous networks, we confine our meta-paths in each task to the ones that start and end both in the target type in the real network cases. For synthetic network tasks, the relation matrices, however, are not homogeneous, for which we only use ITC and ITC-related ensemble algorithms as baselines. Performances under different numbers of seeds in each cluster are tested. Each result is the average of 10 runs.

The accuracy for all 8 tasks is summarized in Tables I to VIII. From the results we can see that, *PathSelClus* performs the best in most of the tasks. Even for a task such as DBLP-T3 where other methods give the best clustering result, *PathSelClus* still gives clustering results among the top. This means, *PathSelClus* can give consistently good results across different tasks in different networks.

Also, by looking at the clustering accuracy trend along with the number of seeds used in each cluster, we can see that using more seeds generally leads to better clustering results.

Table III. Clustering Accuracy for DBLP Tasks: DBLP-T3

#S	Measure	PathSelClus	LP	ITC	LP_voting	LP_soft	ITC_voting	ITC_soft
1	Accuracy	0.8067	**0.9273**	0.5376	0.7091	0.5424	0.4770	0.2358
	NMI	0.6050	**0.7966**	0.5120	0.5870	0.7182	0.3008	0.3416
2	Accuracy	0.9036	**0.9394**	0.5285	0.7333	0.3267	0.5176	0.4085
	NMI	0.7485	**0.8283**	0.5056	0.5986	0.8087	0.3898	0.3464
4	Accuracy	0.9248	**0.9576**	0.7624	0.7636	0.9255	0.6370	0.5485
	NMI	0.7933	0.8841	0.6280	0.6179	**0.9057**	0.4437	0.4634

Table IV. Clustering Accuracy for Yelp Tasks: Yelp-T1

%S	Measure	PathSelClus	LP	ITC	LP_voting	LP_soft	ITC_voting	ITC_soft
1%	Accuracy	**0.5384**	0.3381	0.2619	0.1632	0.1632	0.2564	0.2769
	NMI	**0.5826**	0.0393	0.0042	0.0399	0.0399	0.1907	0.2435
2%	Accuracy	**0.5487**	0.3444	0.2798	0.1713	0.1713	0.3581	0.3790
	NMI	**0.5800**	0.0557	0.0062	0.0567	0.0567	0.2281	0.2734
5%	Accuracy	**0.5989**	0.3732	0.3136	0.1965	0.1965	0.5215	0.5250
	NMI	**0.5796**	0.1004	0.0098	0.0962	0.0962	0.2583	0.2878

Table V. Clustering Accuracy for Yelp Tasks: Yelp-T2

%S	Measure	PathSelClus	LP	ITC	LP_voting	LP_soft	ITC_voting	ITC_soft
1%	Accuracy	**0.7435**	0.1137	0.1758	0.2112	0.2112	0.2430	0.2022
	NMI	**0.6517**	0.0323	0.0178	0.0578	0.0578	0.2308	0.2490
2%	Accuracy	**0.8004**	0.1264	0.1910	0.2202	0.2202	0.2762	0.2792
	NMI	**0.6803**	0.0487	0.0150	0.0801	0.0801	0.2099	0.2907
5%	Accuracy	**0.8125**	0.2653	0.2200	0.2437	0.2437	0.3049	0.3240
	NMI	**0.6894**	0.1111	0.0220	0.1212	0.1212	0.2252	0.2692

Table VI. Clustering Accuracy for Yelp Tasks: Yelp-T3

%S	Measure	PathSelClus	LP	ITC	LP_voting	LP_soft	ITC_voting	ITC_soft
1%	Accuracy	**0.4736**	0.2789	0.1893	0.0682	0.0682	0.2593	0.1775
	NMI	**0.4304**	0.0568	0.0155	0.0626	0.0626	0.1738	0.2065
2%	Accuracy	**0.4597**	0.4008	0.1948	0.0764	0.0764	0.2318	0.2033
	NMI	**0.4359**	0.0910	0.0172	0.0755	0.0755	0.1835	0.1822
5%	Accuracy	0.4393	**0.5351**	0.2233	0.1033	0.1033	0.3337	0.3083
	NMI	**0.4415**	0.1761	0.0194	0.1133	0.1133	0.1793	0.2285

5.2.4. Partial Cluster Seeds. We then test the clustering accuracy when cluster seeds are only available for some of the clusters. We perform this study on DBLP-T3 and Synthetic-T2 using *PathSelClus*, and the results are shown in Figure 4. We can see that even if user guidance is only given to some clusters, those seeds can still be used to improve the clustering accuracy. In general, the fewer the number of clusters with seeds, the worse the clustering accuracy, which is consistent with human intuition.

Note that, label propagation-based methods like LP cannot deal with partial cluster labels. However, in reality it is quite common that users are only familiar with some of the clusters and are only able to give good seeds in those clusters. That is another advantage of *PathSelClus*.

5.3. Efficiency Study

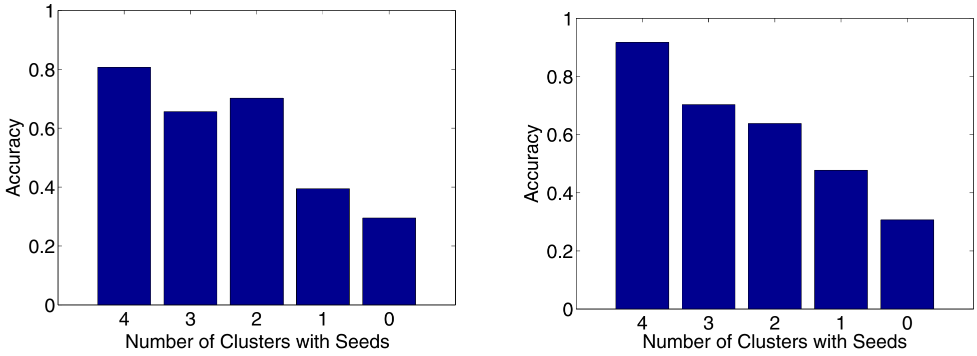
Now, we study the scalability of our algorithm using synthetic datasets, because we can manipulate the size of network flexibly. In Figure 5(a), we keep the size of target objects and the total number of relationships they issued as fixed, and vary the size of feature objects. We can see that the average running time for one iteration of the inner

Table VII. Clustering Accuracy for Synthetic Network Tasks: Synthetic-T1

%S	Measure	PathSelClus	ITC	ITC_voting	ITC_soft
1	Accuracy	**0.9350**	0.2328	0.2879	0.2906
	NMI	**0.7769**	0.1403	0.2231	0.0851
2	Accuracy	**0.9360**	0.3004	0.3482	0.3825
	NMI	**0.7800**	0.0769	0.1253	0.2255
2	Accuracy	**0.9350**	0.3526	0.6417	0.4731
	NMI	**0.7781**	0.1022	0.3285	0.2272

Table VIII. Clustering Accuracy for Synthetic Network Tasks: Synthetic-T2

%S	Measure	PathSelClus	ITC	ITC_voting	ITC_soft
1%	Accuracy	**0.9180**	0.3507	0.3325	0.2912
	NMI	**0.7748**	0.3624	0.1813	0.2652
2%	Accuracy	**0.9170**	0.3978	0.3612	0.3141
	NMI	**0.7728**	0.2923	0.1739	0.2919
5%	Accuracy	**0.9910**	0.6538	0.5373	0.3997
	NMI	**0.9641**	0.5034	0.2001	0.2087



(a) DBLP-T3 with #seeds = 1

(b) Synthetic-T2 with %seeds = 2%

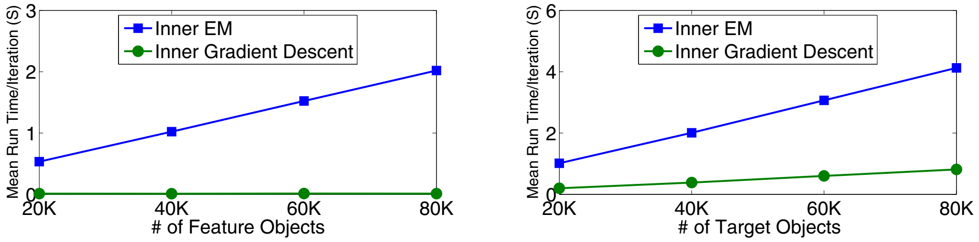
Fig. 4. Clustering accuracy under parietal guidance.

EM algorithm is about linear to the size of the feature objects; and the average running time for one iteration of the inner gradient descent algorithm is almost constant, as it is only linear to the number of relationships in the network. In Figure 5(b), we keep the size of feature objects as fixed, and vary the number of target objects. We keep the average relationships for each target object as constant. From the result we can see that the average running time for one iteration of both the inner EM algorithm and the gradient descent algorithm is linear to the size of target objects, since the number of relationships is also increasing linearly with the size of target objects.

From the efficiency test, we can see that *PathSelClus* is scalable and can be applied to large-scale networks.

5.4. Parameter Study

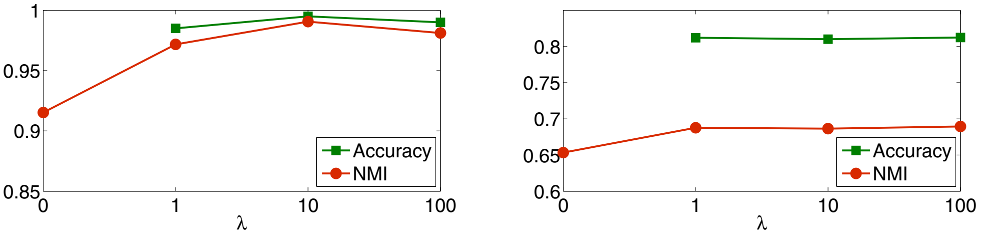
In this section, we study the impact to the performance of our algorithm of the only parameter in the algorithm, λ. We select DBLP-T1 and Yelp-T2 as the test tasks. From the results in Figure 6, we can see that the clustering results are in general not sensitive to the value of λ, as long as it has a positive value. In practice, we set it as 100 for our experiments. Notice that in Figure 6, we do not show the accuracy value when



(a) Varying Feature Object Size

(b) Varying Target Object Size

Fig. 5. Scalability test on synthetic networks.



(a) DBLP-T1: #seeds = 1

(b) Yelp-T2: %seeds = 5%

Fig. 6. Parameter study of λ.

$\lambda = 0$, because when there is no guidance from users, the accuracy cannot be correctly defined.

5.5. Case Studies on Meta-Path Weights

One of the major contributions of *PathSelClus* is that it can select the right meta-paths for a user-guided clustering task. We now show the learned weights of meta-paths for some of the tasks.

In the DBLP-T1 task, the total weight α_m for meta-path $V-P-A-P-V$ is 1576, and the average weight per relationship (a concrete path instance following the meta-path) is 0.0017. The total weight for meta-path $V-P-T-P-V$ is $17,001$, while the average weight per relationship is 0.0003. This means that generally the relationships between two conferences that are connected by an author are more trustable than the ones that are connected by a term, which is consistent with human intuition since many terms can be used in different research areas and authors are typically more focused on confined research topics. However, as there are many more relationships following $V-P-T-P-V$ than following $V-P-A-P-V$, the former provides more information for clustering overall.

In the Yelp network, similar to the DBLP-T1 task, in terms of the average weight for each relationship, meta-path $B-R-U-R-B$ has higher weight than $B-R-T-R-B$; while in terms of total weight, meta-path $B-R-T-R-B$ has higher weight. An interesting phenomenon is that, for the Yelp-T2 task, which tries to cluster restaurants into different categories, the average weight for relationships following $B-R-U-R-B$ is 0.1716, much lower than the value (0.5864) for the Yelp-T3 task, which tries to cluster shopping businesses into finer categories. This simply says that most users actually will try all different kinds of food, therefore they will not serve as a good connection between restaurants as they are in other categories.

In the synthetic networks, for Task 1, that is, Synthetic-T1, the learned quality weights for all three relation matrices are 32,416, 47,620, and 52,892 respectively, and the quality weights for each relationship are 0.5893, 0.8807, and 0.9250, which is consistent with the ground truth of the noise level for each relation matrix, that is, we have successfully assigned lower quality weight to more noisy meta-paths. For Task 2, that is, Synthetic-T2, the learned quality weights for all the three relation matrices are 83,858, 44,353, and 44,532 respectively, and the quality weight for each relationship is 1.3496, 0.7138, and 0.7167, which is consistent with the user guidance, that is, we have successfully assigned the highest quality weight to Relation 1.

6. DISCUSSION

In this article, we have proposed *PathSelClus*, which assigns different weights to different meta-paths for a user-specified clustering task in a heterogeneous information network scenario. In this section, we briefly discuss some interesting issues.

6.1. The Power of Meta-Path Selection

Different meta-paths in heterogeneous networks could be viewed as different sources of information for defining link-based similarity between objects. We first discuss what should be the right level to combine different meta-paths to get the best clustering result, which distinguishes our algorithm from the existing approaches for combining different sources of information to perform a clustering task.

(1) *Combining at the Relation Matrix Level*. The first way to combine information of different meta-paths (or from different sources) is to combine at the relation matrix (feature) level, either by appending the M relation matrices, or by summing up all the relation matrices if all of them are linking the same type. Then traditional feature-based or graph-based semisupervised clustering algorithms can be used to derive the clustering results.
However, there are two limitations of such a level of combination. First, different relation matrices may carry very different scales of values, and there is no easy way to assign proper weights to each relation matrix. Second, the more critical issue is that in many cases, there could be no proper weights at all to linearly combine different sources of information, as they may have completely different semantic meanings. This is why we can see in the experiment section, ITC and LP do not perform well for most of the clustering tasks, which belong to the algorithms that combine information at the relation matrix level.

(2) *Combining at the Clustering Result Level*. Another choice of combining different sources of information is to consider clustering ensemble methods that combine the clustering results and output a consensus clustering result.
However, the major limitation of this level of combination is that if the clustering result obtained by each source is not good enough, the combination of all the clustering results will be not that good. For the motivating example in Example 1.1, we can see that neither $A-O-A$ nor $A-V-A$ provides enough information to cluster authors into 4 groups, and thus the clustering result for each metapath is not good. Their ensemble result turns out to be not good as well. This is also demonstrated in the experiment section, where ITC_soft, ITC_voting, LP_soft, and LP_voting belong to the ensemble algorithms.

(3) *Combining in the Middle*. The third choice is to combine different sources of information in between, which is the option adopted in this article. Instead of combining relation matrices into a single relation matrix, we model the relationships in each relation matrix separately. By looking at the updating formula in Eq. (13) in the M-step, we can see that the clustering result for a target object is determined by

the cluster label of each relationship in each relation matrix. Also, differently from clustering ensemble methods that combine the clustering results at the output level, the learned clustering result Θ will feed back into the modeling of each relation matrix, and to generate a better cluster label for each relationship. It turns out that, in most of cases, this approach is more flexible for combining the information from different sources, and its advantage has been shown in the experiment section.

6.2. Meta-Paths vs. Path Instances

In this article, we only consider the different semantics encoded by different meta-paths. In practice, different concrete paths (path instances) between two objects may also differ from each other, for example, two objects may be linked via a “bridge” or via a “hub,” indicating different meanings. The difference between the two concepts, that is, meta-path and path instance, is similar to the difference between *a source of features* and *a concrete feature* in a vector space. Due to the limited scope, this article only discusses the selection of meta-paths. The theme of selection of path instances at the object level is left to future research.

7. RELATED WORK

Recently, many clustering algorithms have been proposed for networks, for example, spectral clustering-based methods [Luxburg 2007; Shi and Malik 1997]; link-based probabilistic models [Airoldi et al. 2008; Cohn and Chang 2000]; modularity function-based algorithms [Newman 2004; Newman and Girvan 2004]; and density-based algorithms [Wang et al. 2008; Xu et al. 2007] on homogenous networks; and ranking-based algorithms [Sun et al. 2009a, 2009b], nonnegative matrix factorization [Lee and Seung 2000; Wang et al. 2011], spectral clustering-based methods [Long et al. 2006], and probabilistic approaches [Long et al. 2007] on heterogeneous networks. However, while all these clustering methods use the information given in the networks, none considers that different users may have different purposes for clustering, nor do they ask users to help select different information for link-based clustering. In this article, we show that different types of relationships encoded by meta-paths have different semantic meanings in determining the similarity between target objects, and the selection of these meta-paths should be done with user guidance in order to derive user-desired clustering results.

There are several lines of research on how to add user guidance to derive good clustering results, consistent with users’ demands in vector space or networked data.

— *Clustering with constraints*. In Basu et al. [2002, 2004] and Kulis et al. [2005], clustering algorithms that consider constraints either in the form of seeds in each cluster or pairwise constraints such as *must-link* or *cannot-link* are proposed. A probabilistic model with an HMRF (hidden Markov random field) as a hidden layer that models the must-link and cannot-link between objects is proposed to solve the problem [Basu et al. 2004]. This approach can also be extended to graph data with the use of kernels instead of vector-based features [Kulis et al. 2005]. However, these methods assume there is one trustable information source to either define the feature of each object or define the network structure between objects. The goal is to output the clustering result that is consistent with both the similarity defined by the data as well as the user guidance. In this article, we dig further and study which type of information source encoded with meta-paths is more trustable in a heterogeneous network.

— *Semi-supervised learning on graphs*. In Zhu and Ghahramani [2002] and Zhu et al. [2003], algorithms that propagate labels for a small portion of objects into the rest of the network are proposed, which are based on harmonic functions defined between objects using the network structure. Again, these kinds of methods totally trust the given network and determine the best labels of the rest of the nodes according to the cost function defined on the network.

— *Semi-supervised metric learning.* In Bilenko et al. [2004] and Bar-Hillel et al. [2005], algorithms that learn the best distance metric functions according to the constraints for the clustering task are proposed. This line of problem is closer to the meta-path selection problem, but still differs significantly. First, they study features of objects in vector space instead of a network; second, the metric functions should be given in an explicit format, which is very difficult to determine in a network scenario. In this article, we are not to find an explicit metric function that determines the similarity between any two target objects, instead, we model and learn the quality weight for each metapath in the clustering process, which can be viewed as an implicit way to determine the similarity between two target objects.

— *User-guided clustering in relational data.* CrossClus [Yin et al. 2007] deals with another type of guidance from users: the attribute set of the target object type. The algorithm extracts a set of highly relevant attributes in multiple relations connected via linkages defined in the database schema, and then uses the whole attribute set as the feature set to apply traditional vector space-based clustering algorithm. CrossClus works for relational data with complete attributes, but not for purely link-based clustering.

Cluster ensemble [Punera and Ghosh 2008; Strehl et al. 2002] is a method that combines clustering results of different methods or different datasets to a single consensus. Most of these cluster ensemble methods try to find a mean partition given different partitions of target objects. However, in reality, these clusterings may conflict with each other, representing different purposes of clustering tasks, and a consensus does not necessarily lead to a clustering desired by users. In this study, we do not combine clustering results at the output level, but use intermediate clustering results as feedback to adjust the weight of each meta-path, and thus the clustering results and the quality weight for each meta-path can mutually enhance each other.

Our work also differs from traditional feature selection [Guyon and Elisseeff 2003] and the recently emerged semisupervised feature selection [Xu et al. 2010; Zhao and Liu 2007], which focus on vector space features, and do not have an immediate extension to solve our problem. For our metapath selection problem, each meta-path provides a source of features instead of a concrete feature, and we have shown that simple combinations of features from different sources may lead to no good solution.

8. CONCLUSIONS

Link-based clustering for objects in heterogeneous information networks is an important task with many applications. Different from traditional clustering tasks where similarity functions between objects are given and with no ambiguity, objects in heterogeneous networks can be connected via different relationships, encoded by different metapaths. In this article, we integrate the meta-path selection problem with the user-guided clustering problem in heterogeneous networks. An algorithm *PathSelClus* is proposed, which utilizes very limited guidance from users in the form of seeds in some of the clusters, and automatically learns the best weight for each meta-path in the clustering process. The experiments on different tasks on real and synthetic datasets have demonstrated that our algorithm can output the most stable and accurate clustering results compared with the baselines. Also, the learned weights for each meta-path are insightful to explain the hidden similarities between target objects under a particular clustering task.

Exploration of other types of user guidance, such as must-link and cannot-link, in meta-path selection for effective link-based clustering is an interesting topic for future study. More generally, the meta-path selection problem exists in many other mining tasks, such as classification, ranking, and relationship prediction, which requires future research on integrating meta-path selection with all these different mining tasks.

REFERENCES

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. 2008. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res. 9*, 1981–2014.

Banerjee, A., Merugu, S., Dhillon, I. S., and Ghosh, J. 2005. Clustering with Bregman divergences. *J. Mach. Learn. Res. 6*, 1705–1749.

Bar-Hillel, A., Hertz, T., Shental, N., and Weinshall, D. 2005. Learning a Mahalanobis metric from equivalence constraints. *J. Mach. Learn. Res. 6*, 937–965.

Basu, S., Banerjee, A., and Mooney, R. 2002. Semi-supervised clustering by seeding. In *Proceedings of the 19th International Conference on Machine Learning (ICML)*.

Basu, S., Bilenko, M., and Mooney, R. J. 2004. A probabilistic framework for semi-supervised clustering. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 59–68.

Bilenko, M., Basu, S., and Mooney, R. J. 2004. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the 21st International Conference on Machine Learning (ICML).*

Cohn, D. and Chang, H. 2000. Learning to probabilistically identify authoritative documents. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*. 167–174.

Dhillon, I. S., Mallela, S., and Kumar, R. 2003. A divisive information-theoretic feature clustering algorithm for text classification. *J. Mach. Learn. Res. 3*, 1265–1287.

Guyon, I. and Elisseeff, A. 2003. An introduction to variable and feature selection. *J. Mach. Learn. Res. 3*, 1157–1182.

Hofmann, T. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 50–57.

Kulis, B., Basu, S., Dhillon, I., and Mooney, R. 2005. Semi-supervised graph clustering: A kernel approach. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*. 457–464.

Lee, D. D. and Seung, H. S. 2000. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13 (NIPS),* 556–562.

Long, B., Zhang, Z. M., Wu, X., and Yu, P. S. 2006. Spectral clustering for multi-type relational data. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*. 585–592.

Long, B., Zhang, Z. M., and Yu, P. S. 2007. A probabilistic framework for relational clustering. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 470–479.

Luxburg, U. 2007. A tutorial on spectral clustering. *Statist. Comput. 17*, 395–416.

Newman, M. E. J. 2004. Fast algorithm for detecting community structure in networks. *Phys. Rev. E 69*, 6.

Newman, M. E. J. and Girvan, M. 2004. Finding and evaluating community structure in networks. *Phys. Rev. E 69*, 2.

Punera, K. and Ghosh, J. 2008. Consensus-based ensembles of soft clusterings. *Appl. Artif. Intell. 22*, 780–810.

Shi, J. and Malik, J. 1997. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. 22*, 888–905.

Strehl, A., Ghosh, J., and Cardie, C. 2002. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res. 3*, 583–617.

Sun, Y., Han, J., Zhao, P., Yi, N. Z., Cheng, H., and Wu, T. 2009a. Rankclus: Integrating clustering with ranking for heterogeneous information network analysis. In *Proceedings of the 12th International Conference on Extending Database Technology (EDBT)*. 565–576.

Sun, Y., Yu, Y., and Han, J. 2009b. Ranking-based clustering of heterogeneous information networks with star network schema. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 797–806.

Sun, Y., Han, J., Yan, X., Yu, P. S., and Wu, T. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *Proceedings of International Conference on Very Large Data Bases (VLDB)*. 992–1003.

Sun, Y., Norick, B., Han, J., Yan, X., Yu, P. S., and Yu, X. 2012. Integrating meta-path selection with user guided object clustering in heterogeneous information networks. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 1348–1356.

Wang, F., Li, T., Wang, X., Zhu, S., and Ding, C. 2011. Community discovery using nonnegative matrix factorization. *Data Min. Knowl. Discov. 22*, 3, 493–521.

Wang, N., Parthasarathy, S., Tan, K.-L., and Tung, A. K. H. 2008. CSV: Visualizing and mining cohesive subgraphs. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*. 445–458.

Xu, X., Yuruk, N., Feng, Z., and Schweiger, T. A. J. 2007. Scan: A structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 824–833.

Xu, Z., King, I., Lyu, M. R.-T., and Jin, R. 2010. Discriminative semi-supervised feature selection via manifold regularization. *Trans. Neur. Netw. 21*, 1033–1047.

Yin, X., Han, J., and Yu, P. S. 2007. Crossclus: User-guided multirelational clustering. *Data Min. Knowl. Discov. 15*, 3, 321–348.

Zhao, Z. and Liu, H. 2007. Semi-supervised feature selection via spectral analysis. In *Proceedings of the 7th SIAM International Conference on Data Mining (SDM)*.

Zhu, X. and Ghahramani, Z. 2002. Learning from labeled and unlabeled data with label propagation. Tech. rep. CMU-CALD-02-107, Carnegie Mellon University.

Zhu, X., Ghahramani, Z., and Lafferty, J. D. 2003. Semi-Supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference (ICML)*. 912–919.

Received September 2012; accepted December 2012