

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

Yizhou Sun[†] Brandon Norick[†] Jiawei Han[†] Xifeng Yan[‡] Philip S. Yu[§] Xiao Yu[†]

[†] University of Illinois at Urbana-Champaign, Urbana, IL

[‡] University of California at Santa Barbara, Santa Barbara, CA

[§] University of Illinois at Chicago, Chicago, IL

[†]{sun22, bnorick, hanj, xiaoyu1}@illinois.edu [‡]xyan@cs.ucsb.edu [§]psyu@cs.uic.edu

ABSTRACT

Real-world, multiple-typed objects are often interconnected, forming heterogeneous information networks. A major challenge for link-based clustering in such networks is its 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 weights for each meta-path that are 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 are mutually enhancing each other. Our experiments with several clustering tasks in two real networks demonstrate the power of the algorithm in comparison with the baselines.

Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications—*Data mining*

Keywords

Heterogeneous information networks, meta-path selection, user guided clustering

*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-1017362, MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC, and U.S. Air Force Office of Scientific Research MURI award FA9550-08-1-0265. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

KDD’12, August 12–16, 2012, Beijing, China.

Copyright 2012 ACM 978-1-4503-1462-6 /12/08 ...\$15.00.

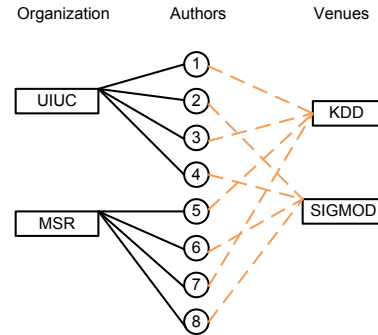


Figure 1: A toy heterogeneous information network containing organizations, authors and venues.

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 (i.e., 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* [21]. These meta-paths imply diverse semantics, and thus clustering on different meta-paths will generate rather different results, as shown below.

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: solid line represents the affiliation relation between author and organization, whereas the dashed one 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 (i.e., colleagues), whereas $A - V - A$ denotes a relation between authors via venues (i.e., publishing in the same venues). A question then raises: *which type 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

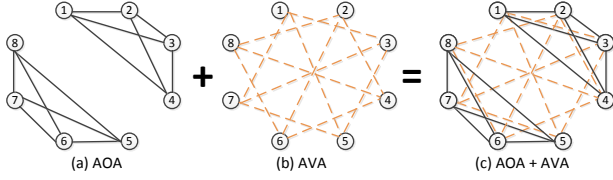


Figure 2: Author connection graphs under different meta-paths.

{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 the 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 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 (*i.e.*, 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 website (*e.g.*, Yelp).

On the other hand, since we are dealing with heterogeneous networks, the previous work on user-guided clustering or semi-supervised learning approaches on (homogeneous) graphs [11, 30, 31] 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 underneath 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, or the one connecting them via text in reviews, or the kNN relation determined by their locations?

In this paper, we integrate the meta-path selection with the user-guided clustering for better clustering 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 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 weights 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-paths quality mutually enhance each other. The experiments with different tasks on two real networks show our algorithm outperforms the baselines. Our contributions are summarized as follows:

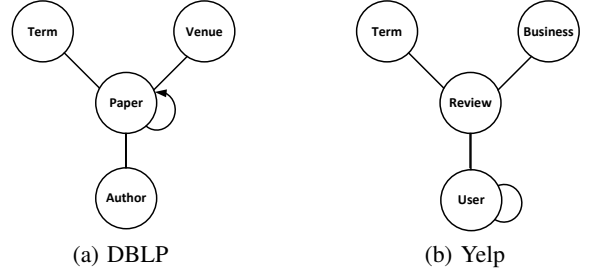


Figure 3: Examples of heterogeneous information networks.

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 each meta-path are learned alternatively and mutually enhance each other; and
3. Experiments on real 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 [22] is an information network with multiple types of objects and/or multiple types of links. Here we introduce two heterogeneous information networks that are used in the experiment section in this paper, which are the DBLP network and the Yelp network.

Example 2.1. (The DBLP bibliographic network¹) 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 relation of “write” and “written by”, between papers and terms by “mention” and “mentioned by”, and between venues and papers by “publish” and “published by”. “Citation” relation between papers can be added further using other data source, such as Google scholar.

Example 2.2. (The Yelp network²) Yelp is a website where users can write reviews for businesses. The Yelp network (see schema in Figure 3(b)) used in this paper contains 4 types of objects, namely business (B), user (U), term (T), and review (R). Links exist between users and reviews by the relation 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 the work [21], 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,

¹<http://www.informatik.uni-trier.de/~ley/db/>

²<http://www.yelp.com/>

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 co-authorship 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 meta-path, e.g., the citation relation $P \rightarrow P$.

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.

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. 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 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, which 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 .
2. The number of clusters, K , and the object seeds for each cluster, say $\mathcal{L}_1, \dots, \mathcal{L}_K$, 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 $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_M$, 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 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, $\theta_i = (\theta_{i1}, \dots, \theta_{iK})$, 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.

A good clustering result is determined by several factors: first, the clustering result should be consistent with the link structure; second, the clustering result should be consistent with the user guidance; third, the importance 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 consideration of all of them.

3.1 Modeling the Relationship Generation

To model the consistency between a clustering result and a relation matrix, 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

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

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 $\pi_{i,m} = (\pi_{i1,m}, \dots, \pi_{i|F_m|,m})$ be the generative probability vector for target object t_i , then each $\pi_{i,m}$ 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 [10], PHITS [7], and RankClus [22], 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 with each other and each relationship generated by t_i is independent with each other, the probability of observing all the relationships between all the target objects and feature objects is the production of the probability of all the relationships following meta-path \mathcal{P}_m :

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

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.

3.2 Modeling the Guidance from Users

Further, we take the 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 θ_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^* , θ_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, θ_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 θ_i given such priors is:

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

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

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

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

In general, a larger λ indicates a higher probability of that θ_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 are connecting 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 a 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., $\Pi_m = \Theta B_m$.

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 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 $\pi_{i,m}$ has a prior of Dirichlet distribution with parameter vector ϕ_i . In this paper, we consider a discrete uniform prior, which is a special case of Dirichlet distribution with parameters as an all-one vector, i.e., $\phi_{i,m} = \mathbf{1}$. 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 $\pi_{i,m}$ for all the target objects t_i , given the observation of relationships $\mathbf{w}_{i,m}$ with relative weight α_m :

$$\alpha_m^* = \arg \max_{\alpha_m} \prod_i P(\pi_{i,m} | \alpha_m \mathbf{w}_{i,m}, \theta_i, B_m) \quad (4)$$

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

$$\pi_{i,m} | \alpha_m \mathbf{w}_{i,m}, \theta_i, B_m \sim \text{Dir}(\alpha_m w_{ij,m} + 1, \dots, \alpha_m w_{i|F_m|,m} + 1) \quad (5)$$

which has the following density function:

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

where $n_{i,m} = \sum_j w_{ij,m}$, 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 j th dimension for the new Dirichlet distribution.
- The larger α_m , the more likely it will generate a $\pi_{i,m}$ with a distribution as the observed relationship distribution, i.e., $\pi_{i,m} \rightarrow \mathbf{w}_{i,m}/n_{i,m}$ 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 π_i with a uniform distribution (which means randomly), i.e., $\pi_{i,m} \rightarrow (1/|F_m|, \dots, 1/|F_m|)$ when $\alpha_m \rightarrow 0$, where $|F_m|$ is the total size of feature objects in meta-path \mathcal{P}_m .

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.

3.4 The Unified Model

By putting all the 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 :

$$P(\{\alpha_m W_m\}_{m=1}^M, \Pi_{1:M}, \Theta | B_{1:M}, \Phi_{1:M}, \lambda) = \prod_i \left(\prod_m P(\alpha_m W_m | \Pi_m, \theta_i, B_m) P(\Pi_m | \Phi_m) \right) P(\theta_i | \lambda) \quad (7)$$

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 , which 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 θ_i for each target object denoting the logarithm of prior density of θ_i :

$$J = \sum_i \left(\sum_m \log P(\pi_{i,m} | \alpha_m \mathbf{w}_{i,m}, \theta_i, B_m) \right) + \sum_k \mathbf{1}_{\{t_i \in \mathcal{L}_k\}} \lambda \log \theta_{ik} \quad (8)$$

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

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

4. THE LEARNING ALGORITHM

In this section, we introduce the learning algorithm, *PathSelClus*, for the model (Eq. (9)) proposed in Section 3. It is a two-step iterative algorithm, where the clustering result Θ and the weights for each meta-path α mutually enhance each other. In the first step, we fix the weight vector α , 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 α .

4.1 Optimize Θ Given α

When α is fixed, the terms only involving α can be discarded in the objective function Eq. (9), which is then reduced to:

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

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:

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

M-step: We have the updating formulas for Θ^t and B_m^t as:

$$\theta_{ik}^t \propto \sum_m \alpha_m \sum_j w_{ij,m} p(z_{ij,m} = k | \Theta^{t-1}, B_m^{t-1}) + \mathbf{1}_{\{t_i \in \mathcal{L}_k\}} \lambda \quad (12)$$

$$\beta_{kj,m}^t \propto \sum_i \sum_j w_{ij,m} p(z_{ij,m} = k | \Theta^{t-1}, B_m^{t-1}) \quad (13)$$

From Eq. (12), we can see that the clustering membership vector θ_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^* , θ_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 Eq. (9) can be reduced to:

$$J_2 = \sum_i \left(\sum_m \left(\sum_j \alpha_m w_{ij,m} \log \pi_{ij,m} + \log \Gamma(\alpha_m n_{i,m} + |F_m|) - \sum_j \log \Gamma(\alpha_m w_{ij,m} + 1) \right) \right) \quad (14)$$

It is easy to check that J_2 is a concave function, which means there is a unique α that maximizes J_2 . We use gradient descent approach to solve the problem, which is an iterative algorithm with the updating formula as: $\alpha_m^t = \alpha_m^{t-1} + \eta_m^t \frac{\partial J_2}{\partial \alpha_m} \Big|_{\alpha_m = \alpha_m^{t-1}}$, where the partial derivative of α_m can be derived as:

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

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 paper, we follow the trick used in non-negative matrix factorization (NMF) [12], and set $\eta_m^t = \frac{\alpha_m^{t-1}}{-\sum_i \sum_j w_{ij,m} \log \pi_{ij,m}}$. By using the above step size, we can get updating formula for α_m as:

$$\alpha_m^t = \alpha_m^{t-1} \frac{\sum_i (\psi(\alpha_m^{t-1} n_{i,m} + |F_m|) n_{i,m} - \sum_j \psi(\alpha_m^{t-1} w_{ij,m} + 1) w_{ij,m})}{-\sum_i \sum_j w_{ij,m} \log \pi_{ij,m}} \quad (15)$$

which guarantees to be a *non-negative* value. Also, by looking at the denominator of the formula, we can see that 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 the 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 α alternatively. The optimization of Θ contains an inner loop of EM-algorithm, and the optimization of α contains another inner loop of gradient descent algorithm. We discuss some details of the algorithm implementation in the following.

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.

Initialization Issues. For the initial value of α , 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 α , if t_i is not labeled, we assign a random clustering vector to θ_i ; while if t_i is labeled as a seed for a cluster k^* , we assign $\theta_i = \mathbf{e}_{k^*}$.

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 $O(t_1(K \sum_m |E_m| + K|T| + K \sum_m |F_m|)) =$

Input: Network: G , Meta-path: $\{\mathcal{P}\}_{m=1}^M$, Number of cluster: K , Object seeds: $\{\mathcal{L}_1, \dots, \mathcal{L}_K\}$, User belief: λ ;
Output: The clustering result Θ ; the weight vector for meta-paths α ;
 Normalize the weight of each relation matrix W_m into $\frac{W_m}{\sum_{ij} W_{ij,m}}$;
 $\alpha = 1$;
repeat
 Initialize Θ^0 and B^0 ;
 repeat
 1. E-step: update $p(z_{ij,m} = k | \Theta^{t-1}, B_m^{t-1})$ by Eq. (11);
 2. M-step: update Θ^t and B_m^t by Eqs. (12) and (13);
 until reaches cluster change threshold;
 $\Theta = \Theta^t$;
 $\alpha^0 = \alpha$;
 repeat
 1. update α^t by Eq. (15);
 until reaches inner α difference threshold;
 $\alpha = \alpha^t$;
until reaches α difference threshold;
 Output Θ and α ;

Algorithm 1: The *PathSelClus* Algorithm.

$O(t_1(K \sum_m |E_m|))$, where $|E_m|$ is the number of non-empty 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 will compare *PathSelClus* with several baselines, and show the effectiveness and efficiency of our algorithm.

5.1 Datasets

In this paper, we use two real information networks for performance test, the DBLP network and the Yelp network. For each network, we design multiple clustering tasks provided with different user guidance, which are 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” [23], 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 been ever 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 sub-network³, which include 6900 businesses, 152327 reviews, and 65888 users. Hierarchical

³http://www.yelp.com/academic_dataset

categories are provided for each business as well, such as “Restaurants”, “Shopping” and so on. For 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 reviews according to two meta-paths: $B - R - U - R - B$ and $B - R - T - R - B$.
- Yelp-T2: We select 6 relatively big sub-categories under the first-level category “Restaurant” (“Sandwiches”, “Thai”, “American (New)”, “Mexican”, “Italian”, and “Chinese”), and cluster 554 businesses with more than one reviews according to the same two meta-paths.
- Yelp-T3: We select 6 relatively big sub-categories 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 reviews according to the same two meta-paths.

5.2 Effectiveness Study

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

5.2.1 Baselines

Three baselines are used in this paper. 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 [4], by replacing Euclidean distance to KL-divergence as used in information theoretic-based clustering algorithms [8, 2]. 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 [31], 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 [18], which can combine soft clustering results into a consensus, which we call *ensemble_soft*. Different 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 into a hard cluster label and then pick 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

Table 1: Clustering Accuracy for DBLP Tasks
(a) 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

(b) 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

(c) 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

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, and 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. In this case, all the three baselines can be used and compared. Performances under different numbers of seeds in each cluster are tested. Each result is the average of 10 runs.

The accuracy for all the 6 tasks for two networks are summarized in Table 1 and Table 2 respectively. From the results we can see that, *PathSelClus* performs the best in most of the tasks. Even for the 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, more seeds generally leads to better clustering results.

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 using *PathSelClus*, which includes 4 clusters, and the results are shown in Fig. 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 number of clusters with seeds, the worse the clustering accuracy, which is consistent with the 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*.

Table 2: Clustering Accuracy for Yelp Tasks
(a) 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

(b) 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

(c) 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

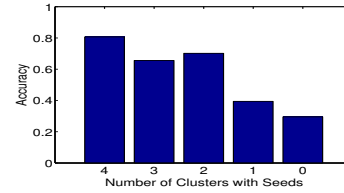


Figure 4: Clustering accuracy under partial guidance: DBLP-T3 with #seeds = 1.

5.3 Efficiency Study

Now, we study the scalability of our algorithm using synthetic datasets, due to that we can manipulate the size of network flexibly. In Fig. 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 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 Fig. 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 very scalable and can be applied to large-scale networks.

5.4 Parameter Study

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

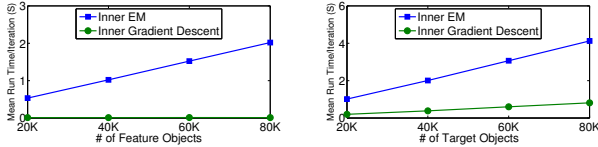


Figure 5: Scalability test on synthetic networks.

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

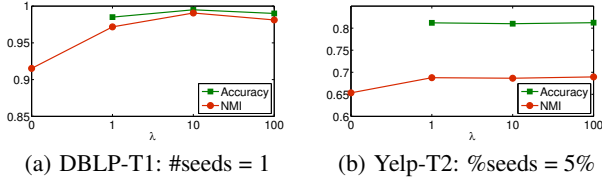


Figure 6: Parameter study of λ .

5.5 Case Study 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 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 17001, 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 much more relationships following $V - P - T - P - V$ than following $V - P - A - P - V$, the former overall provide more information for clustering.

In the Yelp network, similar to DBLP-T1 task, in terms of the average weight for each relationship, meta-path $B - R - U - R - B$ is with higher weight than $B - R - T - U - B$; while in terms of total weight, meta-path $B - R - T - U - B$ is with higher weight. An interesting phenomenon is that, for 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 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 be served as a good connection between restaurants as they are in other categories.

6. DISCUSSION

In this section, we briefly discuss some interesting issues.

1. The Strength 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. There are several ways to handle different meta-paths: (1) to combine them at relation matrix level, such as in baselines ITC and LP; (2) to combine the clustering results at the output level, such as in ensemble baselines; (3) to learn and improve the qual-

ity weights for each meta-path iteratively, such as in *PathSelClus*. Only the third approach is able to select different meta-paths according to different clustering tasks, while the other two can only output an “average” clustering result using all the information. It turns out that, in most cases, the third approach is more flexible to combine information from different sources, and its advantage has been shown in the experiment section.

2. Meta-Paths vs. Path Instances. In this paper, 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, e.g., two objects may be linked via a “bridge” or via a “hub”, indicating different meanings. The difference between the two concepts, i.e., 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 limited scope, this paper only discusses the selection of meta-paths. It is possible to select path instance at the object level, and the concrete method is left for future research.

7. RELATED WORK

Recently, there are many clustering algorithms proposed for networks, such as spectral clustering-based methods [19, 15], link-based probabilistic models [7, 1], modularity function-based algorithms [17, 16], and density-based algorithms [26, 25] on homogenous networks; and ranking-based algorithms [22, 23], non-negative matrix factorization [12, 24], spectral clustering-based methods [13], and probabilistic approaches [14] 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 paper, 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’ demand in vector space or networked data.

Clustering with constraints. In [4, 5, 11], clustering algorithms that consider constraints either in the form of seeds in each cluster or pairwise constraints 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 [5]. This approach can also be extended to graph data with the use of kernels instead of vector-based features [11]. 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 paper, 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 [30, 31], 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, this kind 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 [6, 3], 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 network; second, the metric functions should be given in an explicit format, which is very difficult to determine in a network scenario. In this paper, we are not finding an explicit metric function that determines the similarity between any two target objects, instead, we model and learn the quality weight for each meta-path 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 [28] 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 use 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 [20, 18] 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 [9] and recently emerged semi-supervised feature selection [29, 27], which focus on vector space features, and do not have an immediate extension of solutions to our problem. For our meta-path 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 meta-paths. In this paper, we integrate the meta-path selection problem with the user-guided clustering problem in heterogeneous networks. An algorithm *PathSelClus* that can utilize very limited guidance from users in the form of seeds in some of the clusters and automatically learn the best weights for each meta-path in the clustering process, is proposed. The experiments on different tasks on real 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 very insightful to explain the hidden similarity between target objects under a particular clustering task.

9. REFERENCES

- [1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, June 2008.
- [2] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with bregman divergences. *J. Mach. Learn. Res.*, 6:1705–1749, December 2005.
- [3] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *JMLR*, 6, 2005.
- [4] S. Basu, A. Banerjee, and R. Mooney. Semi-supervised clustering by seeding. In *ICML '02*, 2002.
- [5] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. *KDD '04*, pages 59–68, 2004.
- [6] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *ICML '04*, 2004.
- [7] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. In *ICML '00*, pages 167–174, 2000.
- [8] I. S. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification. *JMLR*, 3:1265–1287, 2003.
- [9] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, Mar. 2003.
- [10] T. Hofmann. Probabilistic latent semantic indexing. *SIGIR '99*, pages 50–57, 1999.
- [11] B. Kulis, S. Basu, I. Dhillon, and R. Mooney. Semi-supervised graph clustering: a kernel approach. *ICML '05*, pages 457–464, 2005.
- [12] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS '00*, pages 556–562, 2000.
- [13] B. Long, Z. (mark) Zhang, X. Wu, and P. S. Yu. Spectral clustering for multi-type relational data. In *ICML '06*, pages 585–592, 2006.
- [14] B. Long, Z. M. Zhang, and P. S. Yu. A probabilistic framework for relational clustering. In *KDD '07*, pages 470–479, 2007.
- [15] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, December 2007.
- [16] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 69(6), 2004.
- [17] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 69(2), 2004.
- [18] K. Punera and J. Ghosh. Consensus-based ensembles of soft clusterings. *Appl. Artif. Intell.*, 22:780–810, August 2008.
- [19] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 1997.
- [20] A. Strehl, J. Ghosh, and C. Cardie. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.
- [21] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *VLDB '11*, 2011.
- [22] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu. Rankclus: integrating clustering with ranking for heterogeneous information network analysis. In *EDBT '09*, pages 565–576, 2009.
- [23] 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.
- [24] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding. Community discovery using nonnegative matrix factorization. *Data Mining and Knowledge Discovery*, 20, 2010.
- [25] N. Wang, S. Parthasarathy, K.-L. Tan, and A. K. H. Tung. Csv: visualizing and mining cohesive subgraphs. In *SIGMOD '08*, pages 445–458, 2008.
- [26] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. Scan: a structural clustering algorithm for networks. In *KDD '07*, pages 824–833, 2007.
- [27] Z. Xu, I. King, M. R.-T. Lyu, and R. Jin. Discriminative semi-supervised feature selection via manifold regularization. *Trans. Neur. Netw.*, 21:1033–1047, July 2010.
- [28] X. Yin, J. Han, and P. S. Yu. Crossclus: user-guided multirelational clustering. *Data Mining and Knowledge Discovery*, 2007.
- [29] Z. Zhao and H. Liu. Semi-supervised feature selection via spectral analysis. In *ICDM '07*, 2007.
- [30] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002.
- [31] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-Supervised learning using gaussian fields and harmonic functions. In *ICML '03*, pages 912–919, 2003.