Ranking-Based Clustering of Heterogeneous Information Networks with Star Network Schema*

Yizhou Sun Yintao Yu Jiawei Han
Department of Computer Science, UIUC
sun22@uiuc.edu yintao@uiuc.edu hanj@cs.uiuc.edu

ABSTRACT

A heterogeneous information network is an information network composed of multiple types of objects. Clustering on such a network may lead to better understanding of both hidden structures of the network and the individual role played by every object in each cluster. However, although clustering on homogeneous networks has been studied over decades, clustering on heterogeneous networks has not been addressed until recently.

A recent study proposed a new algorithm, RankClus, for clustering on bi-typed heterogeneous networks. However, a real-world network may consist of more than two types, and the interactions among multi-typed objects play a key role at disclosing the rich semantics that a network carries. In this paper, we study clustering of multi-typed heterogeneous networks with a *star network schema* and propose a novel algorithm, NetClus, that utilizes links across multi-typed objects to generate high-quality net-clusters. An iterative enhancement method is developed that leads to effective ranking-based clustering in such heterogeneous networks. Our experiments on DBLP data show that NetClus generates more accurate clustering results than the baseline topic model algorithm PLSA and the recently proposed algorithm, RankClus. Further, NetClus generates informative clusters, presenting good ranking and cluster membership information for each attribute object in each net-cluster.

Categories and Subject Descriptors

H.2.8 [**Information Systems Applications**]: Database Applications—*Data Mining*

*The work was supported in part by the U.S. National Science Foundation grants IIS-08-42769 and BDI-05-15813, Office of Naval Research (ONR) grant N00014-08-1-0565, and the Air Force Office of Scientific Research MURI award FA9550-08-1-0265. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

KDD'09, June 28–July 1, 2009, Paris, France.

General Terms

Algorithms

Keywords

heterogeneous information network, clustering

1. INTRODUCTION

Information networks, containing a large number of individual agents or components interacting with each other, are ubiquitous in many applications, *e.g.*, the Internet that consists of a gigantic network of webpages, co-author networks and citation networks extracted from bibliographic data, user networks extracted from email systems, and friendship network extracted from web sites like Facebook[1] and Myspace[2]. Clustering on an information network based on links between objects may give us a grand view of the huge network. For example, communities can be detected by clustering on co-author network [11]. Most current studies [20, 17, 19, 21] on information network are on **homogeneous networks**, *i.e.*, networks consisting of single type of objects, as shown above. However, in reality, objects could be of multiple types, forming a **heterogeneous network**. A recent algorithm RankClus [16] deals with bi-typed heterogeneous networks. Unfortunately, in reality there often exists more than two types of interacting objects in a network. Among them, networks with **star network schema** (called star network) such as bibliographic network centered with papers and tagging network (*e.g.*, http://delicious.com) centered with a tagging event are popular and important. In fact, any n-nary relation set such as records in a relational database can be mapped into a star network, with each relation as the center object and all attribute entities linking to it.

Example 1.1 (Bibliographic Information Network) A bibliographic network consists of rich information about research papers, each *written* by a group of authors, *using* a set of terms, and *published* in a venue (a conference or a journal). Such a bibliographic network is composed of four types of objects: *authors, venues, terms*, and *papers*. Links exist between papers and authors by the relation of "write" and "written by", between papers and terms by the relation of "contain" and "contained in", between papers and venues by the relation of "publish" and "published by". The topological structure of a bibliographic network is shown in the left part of Figure 1, which forms a *star network schema*,

[1]http://www.facebook.com/
[2]http://www.myspace.com/

where paper is a center type and all other types of objects are linked via papers. ■

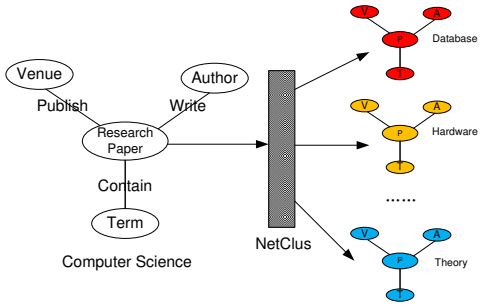


Figure 1: Clustering on A Bibliographic Network

One possible way to cluster a heterogeneous network is to first extract from it a set of homogeneous networks and then apply traditional graph clustering algorithms. However, such an extraction is an information reduction process: some valuable information, *e.g.*, paper title or venue published in, is lost in an extracted co-author network. Further, although clustering co-author network may discover author communities, a research network contains not only authors, but also venues, terms, and papers. It is important to preserve such information by directly clustering on heterogeneous networks, which may lead to generating sub-network clusters carrying rich information. This motivates us to develop *NetClus*, a method that discovers *net-clusters*, *i.e.*, a set of sub-network clusters induced from the original heterogeneous network (Figure 1).

The second weakness of current clustering algorithms is that they do not consider the importance of each object in the network and merely output the cluster label for each object. As a result, clusters are difficult to understand, especially when the size of clusters are large. NetClus not only discovers net-clusters but also gives ranking distribution for each type of objects in each cluster, which makes the cluster so discovered quite meaningful, as shown in the following example.

Example 1.2. (Net-cluster of Database Area) A cluster of the *database area* consists of a set of database authors, conferences, terms, and papers, and can be obtained by NetClus on the bibliographic network extracted from DBLP dataset[3]. NetClus also presents rank scores for authors, conferences, and terms in its own type. With ranking distribution, users can easily grab the important objects in the area. Table 1 shows the top ranked conferences, authors and terms in the area "*database*", generated from a 20-conf. dataset (*i.e.*, a "four-area" dataset) (see details in Sec. 5) using NetClus. ■

Based on the above discussion, in this paper we study the problem of clustering heterogeneous information networks with star network schema and develop a novel clustering algorithm, called NetClus, with the following contributions.

1. A new kind of cluster, *net-cluster*, is proposed for heterogeneous information networks comprised of multiple types of objects. In each cluster, statistical information such as the ranking distribution and membership probability for each object are derived to facilitate users to navigate in the cluster.
2. An effective and efficient algorithm, NetClus, is proposed, that detects net-clusters in a star network with arbitrary number of types, builds ranking-based generative model for each net-cluster and adjusts the membership of target objects according to their posterior probabilities in each net-cluster.
3. Our algorithm is applied to the network extracted from the DBLP dataset, which shows our algorithm can give quite reasonable clustering and ranking results. The clustering accuracy is much higher than the baseline methods.

The rest of paper is organized as follows. Section 2 is an introduction to related work. In Section 3, we formally introduce several concepts related to heterogeneous networks and the clustering problem. In Section 4, we systematically develop the NetClus algorithm. Section 5 is experiment study and Section 6 concludes this study.

2. RELATED WORK

Clustering on networks and graphs has been widely studied in recent years. Clustering on graphs, often called graph partition, aims at partitioning a given graph into a set of subgraphs based on different criteria, such as minimum cut, min-max cut [4] and normalized cut [14]. Spectral clustering [7, 18] provides an efficient method to get graph partitions which is in fact an NP-hard problem. Rather than investigate the global structure like spectral clustering, several density-based methods [19, 21] are proposed to find clusters in networks which utilizes some neighborhood information for each object. These methods are all based on the assumption that the network is homogeneous and the adjacent matrix of the network is already defined.

SimRank [7] is able to calculate pairwise similarity between objects by links of a given network, which could deal with heterogenous network, such as bipartite network. However, when the structure of network becomes more complex such as network with star network schema, SimRank cannot give reasonable similarity measures between objects any more. Also, high time complexity is another issue of SimRank, which prevents it from being applied to large scale networks.

An algorithm called RankClus [16] is newly proposed, which uses a ranking-clustering mutually enhancement methodology to cluster one type of objects in the heterogeneous network. Although the algorithm is efficient comparing to other algorithms that need to calculate pairwise similarity, there are some weaknesses for RankClus: (1) it has not demonstrated the ability to clustering on networks with arbitrary number of types; and (2) the clusters generated by RankClus only contain one type of objects. In contrast, our algorithm can generate net-clusters comprised of objects from multiple types, given any star network.

Other related studies include topic model, such as PLSA [6], which purely uses text information and does not consider link information. Some works such as author-topic model [15] utilizes additional information other than text by designing complex generative models that include additional types of objects. Other works such as [10] intend to optimize a combined objective function with both text and graph constraints. All of these studies are extensions

[3]http://www.informatik.uni-trier.de/∼ley/db/

Conference	Rank Score
SIGMOD	0.315
VLDB	0.306
ICDE	0.194
PODS	0.109
EDBT	0.046
CIKM	0.019
...	...

Author	Rank Score
Michael Stonebraker	0.0063
Surajit Chaudhuri	0.0057
C. Mohan	0.0053
Michael J. Carey	0.0052
David J. DeWitt	0.0051
H. V. Jagadish	0.0043
...	...

Term	Rank Score
database	0.0529
system	0.0322
query	0.0313
data	0.0251
object	0.0138
management	0.0113
...	...

Table 1: Ranking Description for Net-Cluster of Database Research Area

to existing topic model framework, and treat text especially important. In our algorithm, we treat text information just as one common type of objects.

Recently, a different view of clustering on heterogeneous networks [9, 1, 2] appears, which aims at clustering objects from different types simultaneously. Given different cluster number needed for each type of objects, *clusters for each type* are generated by maximizing some objective function. In this paper, net-cluster follows the original network topology and resembles a community that is comprised of *multiple types* of objects.

3. PROBLEM FORMALIZATION

In this section, we define the problem of clustering in heterogeneous information networks and introduce several related concepts and necessary notations.

Definition 1. **Information Network.** Given a set of objects from T types $\mathcal{X} = \{X_t\}_{t=1}^{T}$, where X_t is a set of objects belonging to t_{th} type, a weighted graph $G = \langle V, E, W\rangle$ is called an information network on objects $\mathcal{X}$, if $V = \mathcal{X}$, E is a binary relation on V, and $W : E \to \mathbb{R}^+$ is a weight mapping from an edge $e \in E$ to a real number $w \in \mathbb{R}^+$. Specially, we call such an information network **heterogeneous network** when $T \geq 2$; and **homogeneous network** when $T = 1$.

For convenience, we use X_t to denote both the set of objects belonging to the t_{th} type, and the type name. In the following sections, we will use $W_{x_i x_j}$ to denote the weight of an edge $\langle x_i, x_j\rangle$ in E. $V(G)$, $E(G)$ and $W(G)$ will be denoted as V, E and W of G, if they are not explicitly given.

Definition 2. **Star Network Schema.** An information network $G = \langle V, E, W\rangle$ on $T + 1$ types of objects $\mathcal{X} = \{X_t\}_{t=0}^{T}$ is called with star network schema, if $\forall e = \langle x_i, x_j\rangle \in E, x_i \in X_0 \wedge x_j \in X_t (t \neq 0)$, or vise versa. G is then called a **star network**. Type X_0 is called the center type. X_0 is also called the **target type** and $X_t (t \neq 0)$ are called **attribute types**.

In Example 1.1, paper is the center type in the network and other types of objects only have links to the center type. Many information networks in real applications fall into the class with star network schema. For example, we can build network for tagging website[4], where *tagging event* is the center type, *user, webpage,* and *tag* are linked to tagging events. What's more, each tuple in a database table could be viewed as a center type and each entity attribute in the relation could be viewed as remaining types of objects. Actually, a center object stands for a co-occurrence of different objects, which is able to catch multi-relation instead of binary relation among different objects. In this paper, our algorithm is designed on networks with such topology.

During clustering, center type objects are the objects first be clustered at each iteration, and links to other types of objects are used to help clustering center objects. That is why they are called target type and attribute types.

Definition 3. **Net-cluster.** Given a network G, a net-cluster C is defined as $C = \langle G', p_C\rangle$, where G' is a **sub-network** of G, *i.e.*, $V(G') \subseteq V(G)$, $E(G') \subseteq E(G)$, and $\forall e = \langle x_i, x_j\rangle \in E(G'), W(G')_{x_i x_j} = W(G)_{x_i x_j}$. Function $p_C : V(G') \to [0, 1]$ is defined on $V(G')$, for all $x \in V(G')$, $0 \leq p_C(x) \leq 1$, which denotes the probability that x belongs to cluster C, *i.e.*, $P(x \in C)$.

For convenience, we use $V(C)$ to denote the object set $V(G')$ in network G' and $E(C)$ to denote the edge set $E(G')$. Also, for $x \notin V(G')$, we define $p_C(x) = 0$. In this definition we adopt the idea of soft clustering, which means for each object $x \in V(C)$, it can belong to several clusters with some probability $p_{C_k}(x), k = 1, \ldots, K$ and $\sum_{k=1}^{K} p_{C_k}(x) = 1$. Though actually, for target objects x we restrict $p_C(x)$ as either 0 or 1, and they can belong to merely one cluster. In fact, a net-cluster is a sub-network integrating statistical information for objects. For each net-cluster resembling communities in real world, we argue that it has much simpler structure and can be modeled as a ranking-based generative model. Therefore, every net-cluster is corresponding to a generative model, according to which generative probabilities of every target object in each cluster can be calculated.

Now, we can formalize our clustering problem as: given a heterogeneous information network G, and cluster number K, find K net-clusters $C_1, C_2, \ldots, C_K$, where $\bigcup_{k=1}^{K} V(C_k) = V(G), \bigcup_{k=1}^{K} E(C_k) = E(G)$, and $\forall x \in V(G), \sum_{k=1}^{K} p_{C_k}(x) = 1$, such that target objects within each cluster are nearest to the cluster center under the new K dimensional measure space defined by posterior probabilities.

4. NetClus ALGORITHM

In this section, we introduce an efficient and effective algorithm, NetClus, which is a ranking-based iterative method. The major difficulty that lies in clustering in heterogeneous information network is the definition and calculation of similarity between each pair of objects. The general idea of NetClus is to avoid defining and calculating pairwise similarity between objects but map each target object into a very low dimensional space defined by current clustering result. Then each target object in these clusters will be readjusted based on the new measure. During each iteration, clustering results will be improved under new measure space and the quality of measure will be improved since it is derived from better clusters.

[4] *e.g.*, http://www.delicious.com/

4.1 Framework of NetClus Algorithm

Here, we first introduce the general framework of NetClus, and each part of the algorithm will be explained in detail in the following sections.

The general idea of the NetClus algorithm given cluster number K is composed of the following steps:

- Step 0: Generate initial partitions for target objects and induce initial net-clusters from the original network according to these partitions, *i.e.*, $\{C_k^0\}_{k=1}^K$.
- Step 1: Build ranking-based probabilistic generative model for each net-cluster, *i.e.*, $\{P(x|C_k^t)\}_{k=1}^K$.
- Step 2: Calculate the posterior probabilities for each target object ($p(C_k^t|x)$) and then adjust their cluster assignment according to the new measure defined by the posterior probabilities to each cluster.
- Step 3: Repeat Step 1 and 2 until the cluster does not change significantly, *i.e.*, $\{C_k^*\}_{k=1}^K = \{C_k^t\}_{k=1}^K = \{C_k^{t-1}\}_{k=1}^K$.
- Step 4: Calculate the posterior probabilities for each attribute object ($p(C_k^*|x)$) in each net-cluster.

4.2 Probabilistic Generative Model for Target Objects in a Net-Cluster

According to many studies in real networks [5, 12], preferential attachment and assortative mixing exist in many real networks, which means an object with a higher degree (*i.e.*, high occurrences) has more probability to be attached with an edge, and in some cases higher occurrence objects are more tend to link to each other. As in DBLP dataset, 7.64% of the most productive authors publishes 74.2% of all the papers, among which 56.72% papers are published in merely 8.62% of the biggest venues, which means large size conferences and productive authors are intended to co-appear via papers. We extend the heuristic by using ranking, which denotes the overall importance of an object in a network, instead of degree. The intuition is that degree may not represent global importance of an object well. Examples include: webpage spammed by many low rank webpages linking to it (high-degree but low rank) will not have too much chance to get a link from a real important webpage, and authors publishing many papers in junk conferences will not increase his/her chance to publish a paper in highly ranked conferences. Under this observation, we simplify the network structure by proposing a probabilistic generative model for target objects, where a set of highly ranked attribute objects are more likely to co-appear to generate a center object. To explain this idea, we take bibliographic information network as a concrete example and show how the model works. Bibliographic information network as illustrated in Example 1.1 is formalized as follows.

- Bibliographic Information Network: $G = \langle V, E, W\rangle$.
- Nodes in G: V. In bibliographic network, V is composed of four types of objects: *author* set denoted as A, *conference* set as C, *term* set as T, and *paper* set as D. Suppose the number of distinct objects in each type are $|A|, |C|, |T|$, and $|D|$ respectively, objects in each type are denoted as $A = \{a_1, a_2, \ldots, a_{|A|}\}$, $C = \{c_1, c_2, \ldots, c_{|C|}\}$, $T = \{t_1, t_2, \ldots, t_{|T|}\}$ and $D = \{d_1, d_2, \ldots, d_{|D|}\}$. V is the union of all the objects in all types: $V = A \cup C \cup T \cup D$.
- Edges in G: E and W. In bibliographic network, each paper is written by several authors, published in one conference, and contains several terms in the title. Titles of papers are treated as a bag of terms, in which, the order of terms is unimportant but the number of occurrence of terms is. Therefore, for each paper $d_i, i = 1, 2, \ldots, |D|$, it has three kinds of links, going to three types of attribute objects respectively. For two objects from two arbitrary types, x_i and x_j, if there is a link between them, then edge $\langle x_i, x_j\rangle \in E$. Notice that the graph we consider here is an undirected graph. Also, we use $w_{x_i x_j}$ to denote the weight of the link of edge $\langle x_i, x_j\rangle$, which is defined as follows:

$$w_{x_i x_j} = \begin{cases} 1, \text{ if } x_i(x_j) \in A \cup C \text{ and } x_j(x_i) \in D, \\ \quad \text{and } x_i \text{ has link to } x_j \\ c, \text{ if } x_i(x_j) \in T \text{ and } x_j(x_i) \in D \text{ and } x_i(x_j) \\ \quad \text{appears c times in } x_j(x_i), \\ 0, \text{ otherwise.} \end{cases}$$

In order to simplify the complex network with multiple types of objects, we try to factorize the impact of different types of attribute objects and then model the generative behavior of target objects. The idea of factorizing a network is: we assume that given a network G, the probability to visit objects from different attribute types are independent to each other. Still, the probability to visit an attribute object in G, say author a_i, $p(a_i|G)$ can be decomposed into two parts: $p(a_i|G) = p(A|G) \times p(a_i|A, G)$, where the first part $p(A|G)$ is the overall probability that type of author will be visited in G, and the second part $p(a_i|A, G)$ is the probability that an object a_i will be visited among all the authors in the network G. Generally, given an attribute object x and its type T_x, the probability to visit x in G is defined as in Eq. (1):

$$p(x|G) = p(T_x|G) \times p(x|T_x, G) \tag{1}$$

In practice, $p(T_x|G)$ can be estimated by the proportion of objects in T_x compared with the whole attribute object set $\bigcup T_x$ for all attribute types. Later we will show that the value of $p(T_x|G)$ is not important and can be set to 1. How to generate ranking distribution $p(x|T_x, G)$ for type T_x in a given network G will be addressed in Section 4.4.

Also, we make another independency assumption that within the same type of objects, the probability to visit two different objects jointly is also independent to each other:

$$p(x_i, x_j|T_x, G) = p(x_i|T_x, G) \times p(x_j|T_x, G)$$

where $x_i, x_j \in T_x$ and T_x is some attribute type.

Now, we build the generative model for target objects given the ranking distributions of attribute objects in the network G. Still using bibliographic network as an example, each paper d_i is written by several authors, published in one conference, and comprised of a bag of terms in the title. Therefore, a paper d_i is determined by several attribute objects, say $x_{i1}, x_{i2}, \ldots, x_{in_i}$, where n_i is the number of links d_i has. The probability to generate a paper d_i is equivalent to generating these attribute objects with the occurrence number indicated by the weight of the edge. Under the independency assumptions that we have made, the probability to generate a paper d_i in the network G is defined as follows:

$$\begin{aligned} p(d_i|G) &= \prod_{x \in N_G(d_i)} p(x|G)^{W_{d_i,x}} \\ &= \prod_{x \in N_G(d_i)} p(x|T_x, G)^{W_{d_i,x}} p(T_x|G)^{W_{d_i,x}} \end{aligned}$$

where $N_G(d_i)$ is the neighborhood of object d_i in network G, and T_x is used to denote the type of object x. Intuitively,

a paper is generated in a cluster with high probability, if the conference it is published in, authors writing this paper and terms appeared in the title all have high probability in that cluster.

4.3 Posterior Probability for Target Objects and Attribute Objects

Once we get the generative model for each net-cluster, we can calculate posterior probabilities for each target object. Now the problem becomes that suppose we know the generative probabilities for each target object generated from each cluster $k, k = 1, 2, \ldots, K$, what is the posterior probability that it is generated from cluster k? Here, K is the cluster number given by user. As some target objects may not belong to any of K net-cluster, we will calculate $K+1$ posterior probabilities for each target object instead of K, where the first K posterior probabilities are calculated for each real existing net-clusters $C_1, C_2, \ldots, C_K$, and the last one in fact is calculated for the original network G. Now, the generative model for target objects in G plays a role as background model, and target objects that are not very related to any clusters will have high posterior probability in background model. In this section, we will introduce the method to calculate posterior probabilities for both target objects and attribute objects.

According to the generative model for target objects, the generative probability for a target object d in the target type D in a sub-network $G_k = G(C_k)$ can be calculated according to the *conditional rankings* of attribute types in that sub-network:

$$p(d|G_k) = \prod_{x \in N_{G_k}(d)} p(x|T_x, G_k)^{W_{d,x}} p(T_x|G_k)^{W_{d,x}} \qquad (2)$$

where $N_{G_k}(d)$ denotes for the neighborhood of object d in sub-network G_k. In Eq. (2), in order to avoid zero probabilities in conditional rankings, each conditional ranking should be smoothed using global ranking with smoothing parameter λ_S, before calculating posterior probabilities for target objects:

$$P_S(X|T_X, G_k) = (1 - \lambda_S)P(X|T_X, G_k) + \lambda_S P(X|T_X, G)$$

where λ_S is a parameter that denotes how much we should utilize the ranking distribution from global ranking.

Smoothing [22] is a well-known technology in information retrieval. One of the reasons that smoothing is required in the language model is to deal with the zero probability problem for missing terms in a document. When calculating generative probabilities of target objects using our ranking-based generative model, we meet a similar problem. For example, for a paper in a given net-cluster, it may link to several objects whose ranking score is zero in that cluster. However, if we simply assign the probability of the target object as zero in that cluster, we cannot use other informative objects to decide which cluster this target object is more likely belonging to. In fact, in initial rounds of clustering, objects may be assigned to wrong clusters, if we do not use smoothing technique, they may not have the chance to go back to correct clusters (See the case of $\lambda_S = 0$ in Fig. 4(b)).

Once a clustering is given on the input network G, say $C_1, C_2, \ldots, C_K$, we can calculate the probability for each target object (say paper d_i) simply by Bayesian rule:

$$p(k|d_i) \propto p(d_i|k) \times p(k).$$

where $p(d_i|k)$ is the probability that paper d_i generated from cluster k, and $p(k)$ denotes the relative size of cluster k, *i.e.*, the probability that a paper belongs to cluster k overall. Here, $k = 1, 2, \ldots, K, K+1$. From this formula, we can see that type probability $p(T|G)$ is just a constant for calculating posterior probabilities for target objects and can be neglected.

In order to get the potential cluster size $p(k)$ for each cluster k, we choose cluster size $p(k)$ that maximizes log-likelihood to generate the whole collection of papers and then use the EM algorithm to get the local optimum for $p(k)$.

$$logL = \sum_{i=1}^{|D|} \log(p(di)) = \sum_{i=1}^{|D|} \log[\sum_{k=1}^{K+1} p(di|k)p(k)] \qquad (3)$$

We use the EM algorithm to get $p(k)$ by simply using the following two iterative formulas:

$$p^{(t)}(k|d_i) \propto p(d_i|k)p^{(t)}(k);\ p^{(t+1)}(k) = \sum_{i=1}^{|D|} p^{(t)}(k|d_i)/|D|.$$

Initially, we can set $p^{(0)}(k) = \frac{1}{K+1}$.

When posterior probability is calculated for each target object in each cluster C_k together with the parent cluster C, where $G(C) = G$, each target object d can be represented as a K dimensional vector: $\vec{v}(d) = (p(1|d), p(2|d), \ldots, p(K|d))$. The center for each cluster C_k can be represented using a K dimensional vector as well, which is the mean vector of all the target objects belonging to the cluster under the new measure. Next, we calculate cosine similarity between each target object and each center of cluster, and assign the target object into the cluster with the nearest center. A new sub-network G_k can be induced by current target objects belonging to cluster k. Following the Net-Cluster definition (Definition 3), $p_{C_k}(d) = 1$ if object d is assigned to cluster C_k, 0 otherwise. The adjustment is an iterative process, until target objects do not change their cluster label significantly under the current measure. Notice that, when measuring target objects, we do not use the posterior probability for background model. We make such choices with two reasons: first, the absolute value of posterior probability for background model should not affect the similarity between target objects; second, the sum of the first K posterior probabilities reflects the importance of an object in determining the cluster center.

The posterior probabilities for attribute objects $x \in A \cup C \cup T$ can be calculated as follows:

$$p(k|x) = \sum_{d \in N_G(x)} p(k, d|x) = \sum_{d \in N_G(x)} p(k|d)p(d|x)$$
$$= \sum_{d \in N_G(x)} p(k|d) \frac{1}{|N_G(x)|}$$

It simply says, the probability of a conference belonging to cluster C_k equals to the average posterior probability of papers published in the conference, which is similar for authors. And $p_{C_k}(x)$ in Net-Cluster definition is set to $p(k|x)$.

Example 4.1 (A Running Example of Posterior Change) In Table 2, we select four objects from four types in the DBLP "four-area" dataset to show their posterior probabilities, in four net-clusters and a background model, changing along iterations. Initially, net-clusters are generated from

	Conf: KDD					Author: Michael Stonebraker					Term: Relational					Paper: SimRank[7]				
Iter	DB	DM	ML	IR	BG	DB	DM	ML	IR	BG	DB	DM	ML	IR	BG	DB	DM	ML	IR	BG
0	0.21	0.25	0.19	0.18	0.17	0.32	0.20	0.11	0.20	0.17	0.28	0.20	0.17	0.18	0.17	0.01	0.00	0.87	0.00	0.12
1	0.09	0.32	0.12	0.13	0.34	0.35	0.03	0.01	0.21	0.39	0.31	0.12	0.09	0.15	0.34	0.02	0.00	0.48	0.00	0.50
2	0.02	0.37	0.07	0.10	0.44	0.46	0.00	0.00	0.30	0.24	0.34	0.10	0.10	0.17	0.29	0.01	0.01	0.02	0.00	0.96
5	0.02	0.62	0.07	0.16	0.14	0.74	0.00	0.00	0.19	0.07	0.55	0.13	0.12	0.14	0.06	0.00	0.00	0.00	0.92	0.08
10	0.01	0.89	0.02	0.03	0.04	0.95	0.00	0.00	0.01	0.04	0.68	0.15	0.12	0.02	0.03	0.00	0.32	0.00	0.43	0.25
end	0.01	0.92	0.01	0.03	0.03	0.95	0.00	0.00	0.01	0.03	0.68	0.16	0.11	0.02	0.02	0.00	0.99	0.00	0.00	0.01

Table 2: Illustration of Posterior Change during Iterations for Different Types of Objects

random partitions of papers, each of which is very similar to the original network. Therefore, conditional ranking distributions of each type in each cluster are also very similar to the original ones (background). Thus, posterior probabilities for objects in K initial clusters are similar to each other[5]. However, as similar papers under new measure given by posteriors are grouped together, net-clusters in each area become more and more distinct and objects are gradually assigned with a high posterior probability in the cluster that they should belong to. ∎

4.4 Ranking Distribution for Attribute Objects

Definition 4. **Ranking Distribution and Ranking Function.** A ranking distribution $P(\mathbf{X})$ on a type of objects X is a discrete probability distribution, which satisfies $P(X = x) \geq 0$ $(\forall x \in X)$ and $\sum_{x \in X} P(X = x) = 1$. A function $f_X : G \rightarrow P(\mathbf{X})$ defined on an information network G is called a ranking function on type X, if given an information network G, it can output a ranking distribution $P(\mathbf{X})$ on X.

Ranking is usually used to evaluate the importance or relevance of objects in a collection. For example, PageRank [3] and authority of HITS [8] stand for the static importance of webpages, while the rank of a document to a given query in text retrieval reflects the relevance of the document to that query. Here, we use ranking distribution to represent the importance or visibility of objects within their own type in a given information network G. The higher the rank is, the more possible an object will be visited.

Ranking distributions are quite distinct from each other among different clusters. For example, in computer science area, the ranking distribution of authors from the database area and the system area should be rather different. In the best case, ranking distributions should be orthogonal to each other in different clusters. As we illustrated in Section 4.2, within each cluster, by making independency assumptions between different objects, ranking distributions for each type can be used to build generative models for target objects.

We now introduce two ranking functions using the bibliographic network as an example, and also give some properties of the two ranking functions for a simple 3-typed star network.

1. Simple Ranking

Simple ranking is namely the simple occurrence counting for each object normalized in its own type. Given a network G, ranking distribution for each attribute type of objects is defined as follows:

$$p(x|T_x, G) = \frac{\sum_{y \in N_G(x)} W_{xy}}{\sum_{x' \in T_x} \sum_{y \in N_G(x')} W_{x'y}} \tag{4}$$

[5]Initial absolute posterior prob. to background is sensitive to prior λ_P: the higher λ_P , the larger the value. However, final posterior prob. is not significantly affected by λ_P.

where x is an object from type T_x. For example, in bibliographic network, the rank score for a conference using simple ranking will be proportional to the number of its accepted papers.

PROPERTY 1. *Given a three-typed network with star network schema $G = \langle X \bigcup Y \bigcup Z, E, W \rangle$, where Z is the center type, and $\forall z, N_G(z) = \{x, y\}(x \in X, y \in Y)$, the expected coding error for estimating the joint probability of $P(X, Y)$ by generative model for G under simple ranking $P(X)$ and $P(Y)$ is $I(X, Y)$, where $I(X, Y)$ is the mutual information between X and Y.*

PROOF. $\epsilon = \sum_{x \in X} \sum_{y \in Y} p(x, y)[\log p(x, y) - \log \hat{p}(x, y)]$
$= \sum_{x \in X} \sum_{y \in Y} p(x, y)[\log p(x, y) - \log p(x)p(y)] = I(X, Y)$ □

From the above simple case of network, intuitively for general star network, if a type of attribute objects has a small mutual information with other types of attribute objects, simple ranking is good for it. For example, term type in bibliographic network has small mutual information with authors and conferences in the scale of computer science and database and information system area, and thus could use simple ranking.

2. Authority Ranking

Authority ranking for each object is a ranking function that considers the authority propagation of objects in the network, thus will represent more of the visibility over the whole network. For a general star network G, the propagation of authority score from Type X to Type Y through the center type Z is defined as:

$$P(Y|T_Y, G) = W_{YZ} W_{ZX} P(X|T_X, G) \tag{5}$$

where W_{TZ} and W_{ZX} are the weight matrices between the two types of objects as indexed, and can be normalized when necessary. Generally, authority score of one type of objects could be a combination of scores from different types of objects, *e.g.*, that proposed in [13]. It turns out that the iteration method of calculating ranking distribution is the power method to calculate the primary eigenvector of a square matrix denoting the strength between pairs of objects in that certain type, which can be achieved by selecting a walking path (or a combination of multiple paths) in the network.

PROPERTY 2. *Given a three-typed network with star network schema $G = \langle X \bigcup Y \bigcup Z, E, W \rangle$, where Z is the center type, and $\forall z, N_G(z) = \{x, y\}(x \in X, y \in Y)$, authority ranking $P(X)$ and $P(Y)$ are calculated through Equation 5 iteratively, then estimated joint distribution $\hat{P}(X, Y) = \{\hat{p}(x, y) = P(X = x)P(Y = y), x \in X, y \in Y\}$ equals to the joint distribution represented by one rank matrix $\frac{M}{||M||_1}$, such that $||W_{XZ}W_{ZY} - M||_F$ is minimized.*

PROOF. Let $USV^T = W_{XZ}W_{ZY}$ be SVD of $W_{XZ}W_{ZY}$, and U_1 and V_1 be the first columns of U and V corresponding to the

	C&A	C&T	A&T
Level 1	**0.4564**	0.1389	0.2229
Level 2	**0.3557**	0.1458	0.2502
Level 3	0.2125	0.0065	**0.3968**

Table 3: NMI between Attribute Types in Different Scale of DBLP network

largest singular value σ_1, according to Eckart-Young theorem, $M = \sigma_1 U_1 V_1^T = \min_{\tilde{M}} ||W_{XZ}W_{ZY} - \tilde{M}||$, where $rank(\tilde{M}) = 1$. According to the authority ranking, $P(X) = U_1/||U_1||_1$ and $P(Y) = V_1/||V_1||_1$, thus $M/||M||_1 = \frac{\sigma_1 U_1 V_1^T}{||\sigma_1 U_1 V_1^T||_1} = P(X)P(Y)^T$, where $||M||_1$ is entry-wise 1-norm of M. □

Enlightened by this property holding for the simple network, we can have an intuition that authority ranking is able to catch the largest component structure of a network under the constraints that the relation between objects are recovered by 1-dimensional ranking. As a result, authority ranking should have better performance than simple ranking in most cases. In the DBLP dataset, according to the rules that (1) highly ranked conferences accept many good papers published by many highly ranked authors and (2) highly ranked authors publish many good papers in highly ranked conferences, we determine the iteration equation as:

$$\begin{aligned} P(C|T_C, G) &= W_{CD} D_{DA}^{-1} W_{DA} P(A|T_A G) \\ P(A|T_A, G) &= W_{AD} D_{DC}^{-1} W_{DC} P(C|T_C, G) \end{aligned} \quad (6)$$

where D_{DA} and D_{DC} are the diagonal matrices with the diagonal value equaling to row sum of W_{DA} and W_{DC}. Since all these matrices are sparse, in practice, the rank scores of objects need only be calculated iteratively according to their limited neighbors.

Example 4.2 (Ranking Function Selection in the DBLP Network) Normalized mutual information (NMI) among pairs of two attribute types are calculated in Table 3 in different scales of networks, namely the whole computer science network (level 1), the database and information system network (level 2), and the database network (level 3). Top 1000 objects by occurrence frequency are used in the calculation. If a type has low NMI with all other types, simple ranking is recommended; otherwise, authority ranking is used among types with high NMI. ■

In both ranking functions, prior distributions for a certain type in different clusters can be integrated. Priors for a given type X are given in the form $P_P(X|T_X, k), k = 1, 2, \ldots, K$. An User may give only a few representative objects to serve as priors, like terms and conferences in bibliographic data. First, the prior is propagated in the network in a PageRank way, to propagate scores to objects that are not given in the priors. Then, the propagated prior is linear combined with the ranking functions with parameter $\lambda_P \in [0, 1]$: the bigger the value, the more the final conditional ranking is dependent on prior.

4.5 Algorithm Summary and Time Complexity Analysis

Time complexity of NetClus is composed of the following parts. First, computational complexity for global ranking for attribute objects is $O(t_1|E|)$ and that for global probability calculation for target objects is $O(|E|)$, where $|E|$ is the number of edges in network G and t_1 is the iteration number for ranking. For ranking, at each iteration, each link will be calculated once; and for global probability calculation, a link is still calculated once. Second, time complexity for conditional ranking for attribute objects is $O(t_1|E_k|)$, and for conditional probability for target objects is $O(|E_k|)$ in each cluster k. When adding them together, for all sub-clusters, time complexity for one iteration of clustering should be $O(t_1|E| + |E|)$. Third, time complexity for calculating posterior probability for each target object is $O(t_2(K+1)N)$, where N is the number of target objects, and t_2 is the max iteration number in the EM algorithm. Fourth, cluster adjustment for each target object is $O(K^2N)$. Since for each target object, it has a K dimensional measure, and we have to calculate similarity to K clusters' centers, which are also K-d. Fifth, time complexity for posterior probability for each attribute object is $O(K|E|)$. For each attribute object, each link to target object should be used once to calculate the posterior probability for it. Also, for each attribute type, we have to calculate a K-d measure.

In all, the time complexity for NetClus is $O((t_1+1)|E| + t_3((t_1+1)|E| + t_2(K+1)N + K^2N) + K|E|)$, where t_3 is max iteration number used for clustering adjustment, which can be summarized as $O(c_1|E| + c_2N)$. When the network is very sparse, which is a real situation in most applications, the time complexity is almost linear to the objects in the network.

5. EXPERIMENTS

We now study the effectiveness and accuracy of NetClus and compare it with state-of-the-art algorithms.

5.1 Data Set

We use real data set from DBLP and build bibliographic networks according to Example 1.1. Two networks with different scales will be studied. First, a big data set ("all-area" data set) covers all the conferences, authors, papers and terms from DBLP will be used. Second, we also extract a small data set ("four-area" data set) which contains four areas that are most related to data mining, which are database, data mining, information retrieval and machine learning. Five representative conferences for each area are picked, and all authors have ever published papers on any of the 20 conferences, all these papers and terms appeared in these titles are included in the network. By using the smaller data set, we want to compare the clustering accuracy with several other methods. Also, parameter study and ranking function study will be carried on based on the "four-area" data set.

5.2 Case Study

We first show the ranking distributions in net-clusters we discovered using the "all-area" data set, which is generated by using authority ranking for conferences and authors, setting conference type as priors, and setting the cluster number as 8. We show three net-clusters in Table 4. Also, we can recursively apply NetClus to sub-networks derived from clusters and discover finer net-clusters. Top-5 authors in a finer net-cluster about XML area, which is derived from database sub-network, are shown in Table 5.

5.3 Study on Ranking Functions

In Section 4.4, we proposed two ranking functions, namely simple ranking and authority ranking. Here, we study how low dimensional measure derived from ranking distributions improve clustering and how clustering can improve this new

Rank	DB and IS	Theory	AI
1	SIGMOD	STOC	AAAI
2	VLDB	FOCS	UAI
3	ICDE	SIAM J. Comput.	IJCAI
4	SIGIR	SODA	Artif. Intell.
5	KDD	J. Comput. Syst. Sci.	NIPS

Table 4: Top-5 Conferences in 3 Net-clusters

Rank	Author
1	Serge Abiteboul
2	Victor Vianu
3	Jerome Simeon
4	Michael J. Carey
5	Sophie Cluet

Table 5: Top-5 Authors in "XML" Net-cluster

measure in turn (Figure 2). Here, term is fixed to use simple ranking, and conference and author are set to use authority ranking or simple ranking as two different settings.

First, in order to measure the how different conditional ranking distributions are among clusters, we calculate average KL divergence, which is denoted as $avgD_{KL}(X)$, between each conditional ranking and global ranking for each attribute type X and trace the change of this measure during iterations of clustering. $avgKL(X)$ is defined as:

$$avgD_{KL}(X) = \frac{1}{K}\sum_{k=1}^{K} D_{KL}(P(X|T_X, G_k)||P(X|T_X, G))$$

Second, in order to measure the goodness of measure generated in each round of clustering, we use the compactness, C_f, of target objects under each round of clustering for ranking function f, which is defined as the average ratio between within-cluster similarity and between-cluster similarity using the new measure:

$$C_f = \frac{1}{|D|}\sum_{k=1}^{K}\sum_{i=1}^{|D_k|}\frac{s(d_{ki}, c_k)}{\sum_{k'\neq k} s(d_{ki}, c_{k'})/(K-1)}.$$

Third, we trace the accuracy of clustering results for target objects in each round of iteration, which is defined as:

$$accuracy = \frac{1}{|D|}\sum_{i=1}^{D} P_{true}(\cdot|d_i)\cdot P(\cdot|d_i)$$

However, since $|D|$ is very large even in four-area data set, we manually randomly labeled 100 papers into four clusters and use this paper set to calculate the accuracy.

Fourth, at each iteration of clustering, we calculate the posterior probability for each paper by maximizing the log-likelihood of the whole collection. Here, we also trace the log-likelihood $logL$ along with the clustering iterations, which is defined in Equation 3. From Figure 2, we can see authority ranking is better in every measure than simple ranking.

As we know, in K-means like algorithm, the clustering results are sensitive to initial clustering. We kept 30 times running records and mapped the relation between observable measure of log-likelihood (and compactness) and accuracy into Figure 3 to guide user to pick the best clustering results among several runnings with different initialization. From Figure 3, we can see that linear relation exists among the two measures and accuracy. Also, majority voting among different runnings can be used.

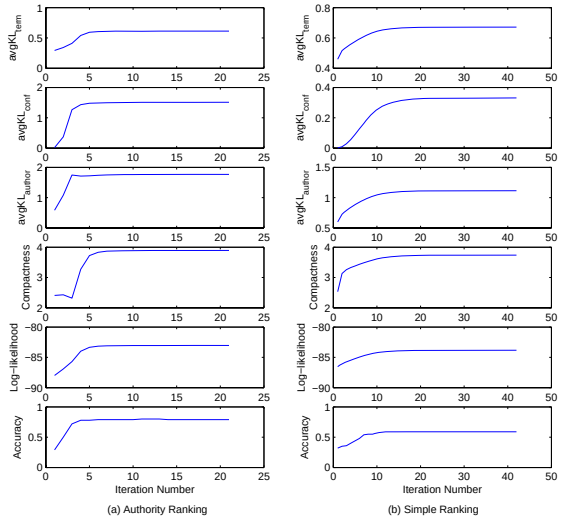


Figure 2: The Change of Average KL Divergence along with the Iteration Number

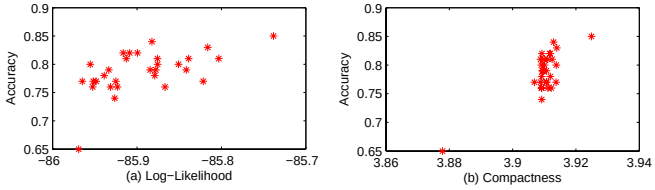


Figure 3: Relation between Log-likelihood / Compactness and Accuracy

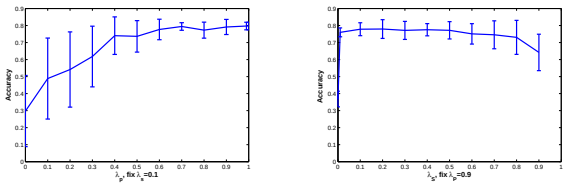
5.4 Study on Parameters

In our algorithm, there are two parameters: prior parameter (λ_P) and smoothing parameter setting (λ_S). We use clustering accuracy for sampled papers to test the impact of different settings of parameters to the algorithm. By fixing one of them, we vary the other one. From Figure 4(a) and 4(b), we find that the larger the prior parameter λ_P, the better the results, while when $\lambda_P > 0.4$, the impact becomes more stable[6]; also, the impact of smoothing parameter is very stable, unless it is not too small (less than 0.1) or too big (bigger than 0.8). The results are based on 20 runnings. Priors given for each of the four areas are around 2 or 3 terms. For example, "database" and "system" are priors for database area, with uniform prior distribution.

5.5 Accuracy Study

In this section, we compare our algorithm with two other algorithms. Since all of them cannot directly applied to heterogeneous network clustering with four types of objects, for each algorithm, we will simplify the network when necessary to make all the algorithms comparable. For PLSA [23], only the term type and paper type in the network are used. No-

[6]Actually, the extremely poor quality when λ_P is very small is partially caused by the improper accuracy measure at those occasions. When the prior is not big enough to attract the papers from the correct cluster, the clusters generated not necessary have the same cluster label with the priors.



(a) Prior: λ_P (b) Smoothing: λ_S

Figure 4: Parameter Study of λ_P and λ_S.

tice that we use the same term prior in both NetClus and PLSA. The accuracy results for papers are in Table 6.

	NetClus (A+C+T+D)	PLSA (T+D)
Accuracy	**0.7705**	0.608

Table 6: Accuracy of Paper Clustering Results

Since RankClus can only cluster conferences, we choose to measure the accuracy of conference cluster. For NetClus, cluster label is obtained according to the largest posterior probability, and NMI [16] is used to measure the accuracy. The results are shown in Table 7, where $d(a) > n$ means we select authors that have more than n publications. Since majority authors only publish a few papers, which contains little information for disclosure of the relationship between two conferences and misleads the algorithm, we run RankClus algorithm by setting different thresholds to select subsets of authors. All the results are based on 20 runnings.

	RankClus $d(a) > 0$	RankClus $d(a) > 5$	RankClus $d(a) > 10$	NetClus $d(a) > 0$
NMI	0.5232	0.8390	0.7573	**0.9753**

Table 7: Accuracy of Conference Clustering Results

6. CONCLUSIONS

In this paper, we address a new clustering problem to detect net-clusters on a special heterogeneous network with star network schema, which aims at splitting the original network into K layers and differs the concept from current clustering methods on heterogeneous networks. A novel ranking-based algorithm called NetClus is proposed to find these clusters. The algorithm makes assumption that within each net-cluster, target objects (*i.e.*, objects from the center type) are generated by a ranking-based probabilistic generative model. Each target object is then mapped into a new low dimensional measure by calculating their posterior probabilities belonging to each net-cluster through their generative models. Our experiments on DBLP data show that NetClus generates more accurate clustering results than the baseline algorithms extended from the topic model and a previous ranking-based algorithm RankClus. Further, NetClus generates more informative clusters, presenting good ranking information and cluster membership for each attribute object in each net-cluster.

In future, we will study how we can automatically set the number of cluster, by which hierarchy tree with arbitrary structure can be detected. Another issue relates to the subspace selection for attribute objects at different scales, which is critical to efficiently and effectively clustering in any complex network.

7. REFERENCES

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

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

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, 1998.

[4] C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *ICDM'01*, pages 107–114. IEEE Computer Society, 2001.

[5] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM'99*, pages 251–262, 1999.

[6] T. Hofmann. Probabilistic latent semantic analysis. In *UAI'99*, pages 289–296, 1999.

[7] G. Jeh and J. Widom. SimRank: a measure of structural-context similarity. In *KDD'02*, pages 538–543. ACM, 2002.

[8] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

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

[10] Q. Mei, D. Zhang, and C. Zhai. A general optimization framework for smoothing language models on graph structures. In *SIGIR'08*, pages 611–618, 2008.

[11] M. E. J. Newman. The structure of scientific collaboration networks. Working Papers 00-07-037, Santa Fe Institute, July 2000.

[12] M. E. J. Newman. Assortative mixing in networks. *Physical Review Letters*, 89(20):208701+, October 2002.

[13] Z. Nie, Y. Zhang, J.-R. Wen, and W.-Y. Ma. Object-level ranking: Bringing order to web objects. In *WWW'05*, pages 567–574. ACM, May 2005.

[14] J. Shi and J. Malik. Normalized cuts and image segmentation. In *CVPR'97*, page 731. IEEE Computer Society, 1997.

[15] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *KDD'04*, pages 306–315, 2004.

[16] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu. Rankclus: Integrating clustering with ranking for heterogenous information network analysis. In *EDBT'09*, 2009.

[17] Y. Tian, R. A. Hankins, and J. M. Patel. Efficient aggregation for graph summarization. In *SIGMOD'08*, pages 567–580, 2008.

[18] U. von Luxburg. A tutorial on spectral clustering. Technical report, Max Planck Institute for Biological Cybernetics, 2006.

[19] 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.

[20] S. White and P. Smyth. A spectral clustering approach to finding communities in graph. In *SDM'05*, 2005.

[21] 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.

[22] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004.

[23] C. Zhai, A. Velivelli, and B. Yu. A cross-collection mixture model for comparative text mining. In *KDD'04*, pages 743–748, 2004.