RankClus: Integrating Clustering with Ranking for Heterogeneous Information Network Analysis[*]

Yizhou Sun[†], Jiawei Han[†], Peixiang Zhao[†], Zhijun Yin[†], Hong Cheng[‡], Tianyi Wu[†]
[†]Department of Computer Science
University of Illinois at Urbana Champaign
{sun22,hanj,pzhao4,zyin3,twu5}@uiuc.edu
[‡]Dept. of Syst. Eng. & Eng. Mgmnt
The Chinese University of Hong Kong
hcheng@se.cuhk.edu.hk

ABSTRACT

As information networks become ubiquitous, extracting knowledge from information networks has become an important task. Both ranking and clustering can provide overall views on information network data, and each has been a hot topic by itself. However, ranking objects globally without considering which clusters they belong to often leads to dumb results, *e.g.*, ranking database and computer architecture conferences together may not make much sense. Similarly, clustering a huge number of objects (*e.g.*, thousands of authors) in one huge cluster without distinction is dull as well.

In this paper, we address the problem of generating clusters for a specified type of objects, as well as ranking information for all types of objects based on these clusters in a multi-typed (*i.e.*, heterogeneous) information network. A novel clustering framework called RankClus is proposed that directly generates clusters integrated with ranking. Based on initial K clusters, ranking is applied separately, which serves as a good measure for each cluster. Then, we use a mixture model to decompose each object into a K-dimensional vector, where each dimension is a component coefficient with respect to a cluster, which is measured by rank distribution. Objects then are reassigned to the nearest cluster under the new measure space to improve clustering. As a result, quality of clustering and ranking are mutually enhanced, which means that the clusters are getting more accurate and the ranking is getting more meaningful. Such a progressive refinement process iterates until little change can be made. Our experiment results show that RankClus can generate more accurate clusters and in a more efficient way than the state-of-the-art link-based clustering methods. Moreover, the clustering results with ranks can provide more informative views of data compared with traditional clustering.

1. INTRODUCTION

In many applications, there exist a large number of individual agents or components interacting with a specific set of components, forming large, interconnected, and sophisticated networks. We call such interconnected networks as *information networks*, with examples including the Internet, highway networks [10], electrical power grids, research collaboration networks [6], public health systems, biological networks [14], and so on. Clearly, information networks are ubiquitous and form a critical component of modern information infrastructure. Among them, heterogeneous network is a special type of network that contains objects of multiple types.

A great many analytical techniques have been proposed toward a better understanding of information networks and their properties, among which are two prominent ones: *ranking* and *clustering*. On one hand, *ranking* evaluates objects of information networks based on some *ranking function* that mathematically demonstrates characteristics of objects. With such functions, any two objects of the same type can be compared, either qualitatively or quantitatively, in a partial order. PageRank [2] and HITS [11], among others, are perhaps the most renowned ranking algorithms over information networks. On the other hand, *clustering* groups objects based on a certain proximity measure so that similar objects are in the same cluster, whereas dissimilar ones are in different clusters. After all, as two fundamental analytical tools, ranking and clustering demonstrate overall views of information networks, and hence be widely applied in different information network settings.

Clustering and ranking are often regarded as *orthogonal* techniques, each of which is applied separately to information network analysis. However, applying either of them over information networks often leads to incomplete, or sometimes rather biased, analytical results. For instance, ranking objects over the global information networks without considering which clusters they belong to often leads to dumb results, *e.g.*, ranking database and computer architecture conferences and authors together may not make much sense; alternatively, clustering a large number of objects (*e.g.*, thousands of authors) in one cluster without distinction is dull as well. However, combining both functions together may lead to more comprehensible results, as shown below.

[*]The work was supported in part by the U.S. National Science Foundation grants IIS-08-42769 and BDI-05-15813, NASA grant NNX08AC35A and Office of Naval Research (ONR) grant N00014-08-1-0565. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

EDBT 2009, March 24–26, 2009, Saint Petersburg, Russia.
Copyright 2009 ACM 978-1-60558-422-5/09/0003 ...$5.00

Example 1.1 (Ranking without clustering) Consider a set of conferences from two areas of (1) DB/DM (*i.e.*, *Database and Data Mining*) and HW/CA (*i.e.*, *Hardware and Computer Architecture*), each having 10 conferences, as shown in Table 1. Then we choose 100 authors in each area from DBLP [4]. With the ranking function specified in Sec. 4.2, our ranking-only algorithm gives top-10 ranked results (Table 2). Clearly, the results are rather dumb (because of the mixture of the areas) and are biased towards (*i.e.*, ranked higher for) the HW/CA area. What is more, such dull or biased ranking result is caused not by the specific ranking function we chose, but by the inherent incomparability between the two areas. ■

Table 1: A set of conferences from two research areas

DB/DM	{SIGMOD, VLDB, PODS, ICDE, ICDT, KDD, ICDM, CIKM, PAKDD, PKDD}
HW/CA	{ASPLOS, ISCA, DAC, MICRO, ICCAD, HPCA, ISLPED, CODES, DATE, VTS }

Table 2: Top-10 ranked conferences and authors in the mixed conference set

Rank	Conf.	Rank	Authors
1	DAC	1	Alberto L. Sangiovanni-Vincentelli
2	ICCAD	2	Robert K. Brayton
3	DATE	3	Massoud Pedram
4	ISLPED	4	Miodrag Potkonjak
5	VTS	5	Andrew B. Kahng
6	CODES	6	Kwang-Ting Cheng
7	ISCA	7	Lawrence T. Pileggi
8	VLDB	8	David Blaauw
9	SIGMOD	9	Jason Cong
10	ICDE	10	D. F. Wong

Table 3: Top-10 ranked conferences and authors in DB/DM set

Rank	Conf.	Rank	Authors
1	VLDB	1	H. V. Jagadish
2	SIGMOD	2	Surajit Chaudhuri
3	ICDE	3	Divesh Srivastava
4	PODS	4	Michael Stonebraker
5	KDD	5	Hector Garcia-Molina
6	CIKM	6	Jeffrey F. Naughton
7	ICDM	7	David J. DeWitt
8	PAKDD	8	Jiawei Han
9	ICDT	9	Rakesh Agrawal
10	PKDD	10	Raghu Ramakrishnan

Example 1.2 (Ranking based on good clusters) Still consider the data set introduced in Ex. 1.1, this time we picked 10 conferences in the DB/DM area and rank them as well as the authors relative to this conference cluster. The ranking results are shown in Table 3. ■

Ex. 1.2 shows that good cluster indeed enhances ranking results. Moreover, assigning ranks to objects often leads to better understanding of each cluster. Obviously, good clusters promote good ranking, but how to get good clusters? A straightforward way is to first evaluate similarity between objects using a link-based method, such as SimRank [9], and then apply graph clustering methods [15, 12] or the like to generate clusters. However, to evaluate similarity between objects in an arbitrary multi-typed information network is a difficult and time-consuming task. Instead, we propose RankClus that explores rank distribution for each cluster to improve clustering, and the basic idea is as follows. Based on initial K clusters, ranking is applied separately, which serves as a good measure for each cluster. Then, a mixture model is used to decompose each object into a K-dimensional vector, where each dimension is a component coefficient with respect to a cluster, which is measured by rank distribution. Objects then are reassigned to the nearest cluster under the new measure space. As a result, the quality of clustering is improved. What is more, ranking results can thus be enhanced further by these high quality clusters. In all, instead of combining ranking and clustering in a two stage procedure like facet ranking [3, 18], the quality of clustering and ranking can be mutually enhanced in RankClus.

In this paper, we propose RankClus, a novel framework that smoothly integrates clustering and ranking. Given a user-specified target type, our algorithm directly generates clusters for the target objects from target type as well as rank information for all the objects based on these clusters in the network. Our study shows that RankClus can generate more accurate clusters than the state-of-the-art link-based clustering method in a more effective and comprehensive way. Moreover, the clustering results with ranks can provide more informative views of data. The main contributions of our paper are as follows.

1. We propose a general framework in which ranking and clustering are successfully combined to analyze information networks. To our best knowledge, our work is the first to advocate making use of both ranking and clustering simultaneously for comprehensive and meaningful analysis of large information networks.
2. We formally study how ranking and clustering can mutually reinforce each other in information network analysis. A novel algorithm called RankClus is proposed and its correctness and effectiveness are verified.
3. We perform a thorough experimental study on both synthetic and real datasets in comparison with the state-of-the-art algorithms, and the experimental results demonstrate the power of RankClus.

The rest of paper is organized as follows. Section 2 is on related work. In Section 3, we define and illustrate several important concepts to be used in subsequent sections. In Section 4, we use the DBLP data as an example of a bi-type information network, and define two ranking functions on it. In Section 5, we propose the RankClus algorithm, taking bi-type information network as an example. Section 6 is a systematic experimental analysis on both synthetic and real datasets. We discuss our methodology in Section 7 and conclude our study in Section 8.

2. RELATED WORK

In information network analysis, two most important ranking algorithms are PageRank [2] and HITS [11], both of which are successfully applied to the Internet search. PageRank is a link analysis algorithm that assigns a numerical weight to each object of the information network, with the purpose of "measuring" its relative importance within the object set. On the other hand, HITS ranks objects based on two scores: *authority* and *hub*. Authority estimates the value of the content of the object, whereas hub measures the value of its links to other objects. Both PageRank and HITS are evaluating the static quality of objects in information network, which is similar to the intrinsic meaning of our ranking methods. However, both PageRank and HITS are designed on the network of web pages, which is a directed homogeneous network, and the weight of the edge is binary. PopRank [13] aims at ranking popularity of web objects. They have considered the role difference of different web pages, and thus turn web pages into a heterogeneous network. They trained the propagation factor between different types of objects according to partial ranks given by experts. Different from their setting, we will calculate the rank for each type of objects seperately (*i.e.*, we do not compare ranks of two objects belonging to different types), rather than consider them in a unified framework. J. E. Hirsch [8] proposed h index originally in the area of physics for characterizing the scientific output of a researcher, which is defined as the number of papers with citation number higher or equal to h. Extensions work [16] shows that it also can work well in computer science area. However, h-index will assign an integer value h to papers, authors, and publication forums, while our work requires that rank sores can be viewed as a rank distribution and thus can serve as a good measure for clustering. What is more, since there are only very limited citation information in DBLP, ranking methods demanding citation cannot work in such kind of data. Instead of proposing a totally new strategy for ranking, we aim at finding empirical rules in the specific area of DBLP data set, and providing ranking function based on these rules, which works well for the specific case. The real novelty lies in our framework is that it tightly integrates ranking and clustering and thus offers informative summary for heterogeneous network such as the DBLP data.

Clustering is another way to summarize information network and discover the underlying structures, which partitions the objects of an information network into subsets (clusters) so that objects in each subset share some common trait. In clustering, proximity between objects is often defined for the purpose of grouping "similar" objects into one cluster, while partitioning dissimilar ones far apart. Spectral graph clustering [15, 12] is state-of-the-art method to do clustering on the homogeneous network. However for heterogeneous network, adjacency matrix of the same type objects are not explicit existing. Therefore, similarity extraction methods such as SimRank [9] should be applied first, which is an iterative PageRank-like method for computing structural similarity between objects. However, the time cost for SimRank is very high, and other methods such as LinkClus [17] have addressed this issue. Without calculating the pairwise similarity between two objects of the same type, RankClus uses conditional ranking as the measure of clusters, and only needs to calculate the distances between each object and the cluster center.

In web search, there exists an idea of facet ranking [18, 3], which clusters the returned results for each query into different categories, to help users to better retrieve the relevant documents. A commercial website that illustrates the idea is "vivisimo.com" [1]. It may seem that facet ranking also integrates ranking with clustering, however, our work is of totally different idea. First, the goal of facet ranking is to help user to better organize the results. The meaning of ranking here is the relevance to the query. RankClus aims at finding higher quality and more informative clusters for target objects with rank information integrated in an information network. Second, facet ranking is a two-stage methodology. In the first stage, relevant results are collected according to the relevance to the query, and then clustering is applied on the collection of returned documents. RankClus integrates ranking and clustering tightly, which are mutually improved during the iterations.

3. PROBLEM DEFINITION

Among many information networks, bi-type information network is popular in many applications. For example, conference-author network in bibliographic database, movie-user network in online movie database, and newsgroup-author network in newsgroup database. In this paper, we use bi-type network as an example to illustrate RankClus algorithm. Accordingly, most concepts introduced are based on bi-type information network.

Definition 1. ***Bi-type Information Network.*** *Given two types of object sets X and Y, where $X = \{x_1, x_2, \ldots, x_m\}$, and $Y = \{y_1, y_2, \ldots, y_n\}$, graph $G = \langle V, E\rangle$ is called a bi-type information network on types X and Y , if $V(G) = X \cup Y$ and $E(G) = \{\langle o_i, o_j\rangle\}$, where $o_i, o_j \in X \cup Y$.*

Let $W_{(m+n)\times(m+n)} = \{w_{o_io_j}\}$ be the adjacency matrix of links, where $w_{o_io_j}$ equals to the weight of link $\langle o_i, o_j\rangle$, which is the observation number of the link, we thus use $G = \langle\{X \cup Y\}, W\rangle$ to denote this bi-type information network. In the following, we use X and Y denoting both the object set and their type name. For convenience, we decompose the link matrix into four blocks: W_{XX}, W_{XY}, W_{YX} and W_{YY}, each denoting a sub-network of objects between types of the subscripts. W thus can be written as:

$$W = \begin{pmatrix} W_{XX} & W_{XY} \\ W_{YX} & W_{YY} \end{pmatrix}$$

Definition 2. ***Ranking Function.*** *Given a bi-type network $G = \langle\{X \cup Y\}, W\rangle$, if a function $f : G \to (\vec{r}_X, \vec{r}_Y)$ gives rank score for each object in type X and type Y, where*

$$\forall x \in X, \vec{r}_X(x) \geq 0, \sum_{x\in X} \vec{r}_X(x) = 1, \text{ and}$$
$$\forall y \in Y, \vec{r}_Y(y) \geq 0, \sum_{y\in Y} \vec{r}_Y(y) = 1,$$

we call f a ranking function on network G.

The aim of ranking in information network is to give different importance weights to different objects. Thus, users can quickly navigate to important objects. For example, PageRank is a ranking function defined on the Web, which

[1] http://vivisimo.com

is a single-type information network with web pages as its objects. For the bi-type information network defined in the DBLP data, we will provide two ranking functions in Section 4.

For a given cluster number K, clustering is to give a cluster label from 1 to K for each object in the *target type* X. We use X_k to denote the object set of cluster k, and use X' to denote an arbitrary cluster. In most bi-type networks, the two types of objects could be rather asymmetric in cardinality. For example, in DBLP, the number of authors is around 500,000, and the number of conferences is only around 4,000. In our method, we treat the type (of objects) that contains less number of distinct values as **target type** in the information network, whereas the other as **attribute type**. Clustering is only applied to the target type objects in order to generate less number but more meaningful clusters; whereas the attribute type objects only help the clustering. Taking DBLP as an example, we recommend to only consider conference as target type for clustering because (1) we only need small number for clusters, which has the intrinsic meaning of research area, and (2) authors' rank score in each conference cluster has already offered enough information.

As shown in Section 1, ranking of objects without considering which clusters they belong to often leads to dumb results. Therefore, we introduce the concept of *conditional rank*, which is the rank based on a specific cluster.

Definition 3. ***Conditional rank** and **within-cluster rank.*** *Given target type X, and a cluster $X' \subseteq X$, sub-network $G' = \langle \{X' \cup Y\}, W' \rangle$ is defined as a vertex induced graph of G by sub vertex set $X' \cup Y$. Conditional rank over Y, denoted as $\vec{r}_{Y|X'}$, and within-cluster rank over X', denoted as $\vec{r}_{X'|X'}$, are defined by the ranking function f on the sub-network G': $(\vec{r}_{X'|X'}, \vec{r}_{Y|X'}) = f(G')$. Conditional rank over X, denoted as $\vec{r}_{X|X'}$, is defined as the propagation score of $\vec{r}_{Y|X'}$ over network G:*

$$\vec{r}_{X|X'}(x) = \frac{\sum_{j=1}^{n} W_{XY}(x,j)\vec{r}_{Y|X'}(j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} W_{XY}(i,j)\vec{r}_{Y|X'}(j)}.$$

In this definition, conditional rank over Y and within-cluster rank over X' are straightforward, which are the application of ranking function on the sub-network G' induced by cluster X'. Conditional rank over whole set of objects in X is more complex, since not every object in X is in the sub-network G'. The idea behind the concept is that when a cluster X' is given, and conditional rank over Y, which is $\vec{r}_{Y|X'}$, is calculated, the conditional rank over X relative to cluster X' can be determined according to current rank of Y. For example, once DB/DM conference cluster is given, we can then get authors' conditional rank on DB/DM cluster, and whether a conference's conditional rank score relative to DB/DM cluster is high is determined by whether many of the authors in the conference are highly ranked in DB/DM area. The detailed calculation and explanation of ranking are provided in Section 4, based on two concrete ranking functions.

Based on these definitions, our goal of this paper can be summarized as follows: given a bi-type network $G = \langle \{X \cup Y\}, W \rangle$, the target type X, and a specified cluster number K, our goal is to generate K clusters $\{X_k\}$ on X, as well as the within-cluster rank for type X and conditional rank for type Y to each cluster, *i.e.*, $\vec{r}_{X|X_k}$ and $\vec{r}_{Y|X_k}$, $k = 1, 2, \ldots, K$.

4. RANKING FUNCTION

Ranking can give people an overall view of a certain set of objects, which is beneficial for people to grasp the most important information in a short time. More importantly, in this paper, conditional ranks of attribute types are served as features for each cluster, and each object in target type can be considered as a mixture model over these rank distributions, and the component coefficients can be used to improve clustering. In this section, we propose two ranking functions that could be used frequently in bi-type network similar to conference-author network. In bibliographic network, consider the bi-type information network composed of conferences and authors. Let X be the type of conference, Y be the type of author, and specify conference as the target type for clustering. According to the publication relationship between conferences and authors, we define the *link matrix* W_{XY} as:

$$W_{XY}(i,j) = p_{ij}, \text{ for } i = 1, 2, \ldots, m; j = 1, 2, \ldots, n$$

where p_{ij} is the number of papers that author j published in conference i, or equally, the number of papers in conference i that are published by author j. According to the co-author relationship between authors, we define the matrix W_{YY} as:

$$W_{YY}(i,j) = a_{ij}, \text{ for } i = 1, 2, \ldots, m; j = 1, 2, \ldots, n$$

where a_{ij} is the number of papers that author i and author j co-authored. The link matrix denoting the relationship between authors and conferences W_{YX} is equal to W_{XY}^T, as the relationship between authors and conferences is symmetric, and $W_{XX} = 0$ as there are no direct links between conferences. Based on this conference-author network, we define two ranking functions: *Simple Ranking* and *Authority Ranking.*

4.1 Simple Ranking

The simplest ranking of conferences and authors is based on the number of publications, which is proportional to the numbers of papers accepted by a conference or published by an author.

Given the information network $G = \langle \{X \cup Y\}, W \rangle$, simple ranking generates the ranking score of type X and type Y as follows:

$$\begin{cases} \vec{r}_X(x) = \frac{\sum_{j=1}^{n} W_{XY}(x,j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} W_{XY}(i,j)} \\ \vec{r}_Y(y) = \frac{\sum_{i=1}^{n} W_{XY}(i,y)}{\sum_{i=1}^{m}\sum_{j=1}^{n} W_{XY}(i,j)} \end{cases} \quad (1)$$

The time complexity of Simple Ranking is $O(|E|)$, where $|E|$ is the number of links.

Obviously, simple ranking is only a normalized weighted degree of each object, which considers every link equally important. In this ranking, authors publishing more papers will have higher ranking score, even these papers are all in junk conferences. In fact, simple ranking evaluate importance of each object according to their immediate neighborhoods.

4.2 Authority Ranking

A more useful ranking we propose here is authority ranking function, which gives an object higher ranking score if it has more authority. Ranking authority merely with publication information seems impossible at first, as citation information could be unavailable or incomplete (such as in the DBLP data, where there is no citation information imported from Citeseer, ACM Digital Library, or Google Scholars). However, two simple empirical rules give us the first clues.

- Rule 1: Highly ranked authors publish *many* papers in highly ranked conferences.
- Rule 2: Highly ranked conferences attract *many* papers from *many* highly ranked authors.

Notice that these empirical rules are domain dependent and are usually given by the domain experts who know both the field and the data set well[2].

From the above heuristics, we define the ranking score of authors and conferences according to each other as follows.

According to Rule 1, each author's score is determined by the number of papers and their publication forums,

$$\vec{r}_Y(j) = \sum_{i=1}^{m} W_{YX}(j,i)\vec{r}_X(i). \tag{2}$$

When author j publishes more papers, there are more nonzero and high weighted $W_{YX}(j,i)$, and when the author publishes papers in a higher ranked conference i, which means a higher $\vec{r}_X(i)$, the score of author j will be higher. At the end of each step, $\vec{r}_Y(j)$ is normalized by

$$\vec{r}_Y(j) \leftarrow \frac{\vec{r}_Y(j)}{\sum_{j'=1}^{n} \vec{r}_Y(j')},$$

According to Rule 2, the score of each conference is determined by the quantity and quality of papers in the conference, which is measured by their authors' ranking scores,

$$\vec{r}_X(i) = \sum_{j=1}^{n} W_{XY}(i,j)\vec{r}_Y(j). \tag{3}$$

When there are more papers appearing in conference i, there are more non-zero and high weighted $W_{XY}(i,j)$; if the papers are published by higher ranked author j, the rank score for j, which is $\vec{r}_Y(j)$, is higher, and thus the higher score the conference i will get. The score vector is then normalized:

$$\vec{r}_X(i) \leftarrow \frac{\vec{r}_X(i)}{\sum_{i'=1}^{m} \vec{r}_X(i')},$$

Notice that the normalization will not change the ranking position of an object, but it gives a relative importance score to each object. The two formulas can be rewritten using the matrix form:

$$\begin{cases} \vec{r}_X = \dfrac{W_{XY}\vec{r}_Y}{\|W_{XY}\vec{r}_Y\|} \\ \vec{r}_Y = \dfrac{W_{YX}\vec{r}_X}{\|W_{YX}\vec{r}_X\|} \end{cases} \tag{4}$$

THEOREM 1. *The solution to $\vec{r}_X$ and $\vec{r}_Y$ given by the iteration formula is the primary eigenvector of $W_{XY}W_{YX}$ and $W_{YX}W_{XY}$ respectively.*

PROOF. Combining Eqs. (2) and (3), we get

$$\vec{r}_X = \frac{W_{XY}\vec{r}_Y}{\|W_{XY}\vec{r}_Y\|} = \frac{W_{XY}\frac{W_{YX}\vec{r}_X}{\|W_{YX}\vec{r}_X\|}}{\|W_{XY}\frac{W_{YX}\vec{r}_X}{\|W_{YX}\vec{r}_X\|}\|} = \frac{W_{XY}W_{YX}\vec{r}_X}{\|W_{XY}W_{YX}\vec{r}_X\|}$$

Thus, $\vec{r}_X$ is the eigenvector of $W_{XY}W_{YX}$. The iterative method is the power method [5] to calculate the eigenvector, which is the primary eigenvector. Similarly, $\vec{r}_Y$ is the primary eigenvector of $W_{YX}W_{XY}$. □

When considering the co-author information, the scoring function can be further refined by a third rule:

- Rule 3: The rank of an author is enhanced if he or she coauthors with many authors or many highly ranked authors.

Using this new rule, we can revise Eqs. (2) as

$$\vec{r}_Y(i) = \alpha \sum_{j=1}^{m} W_{YX}(i,j)\vec{r}_X(j) + (1-\alpha)\sum_{j=1}^{n} W_{YY}(i,j)\vec{r}_Y(j). \tag{5}$$

where parameter $\alpha \in [0,1]$ determines how much weight to put on each factor based on one's belief.

Similarly, we can prove that $\vec{r}_Y$ should be the primary eigenvector of $\alpha W_{YX}W_{XY} + (1-\alpha)W_{YY}$, and $\vec{r}_X$ should be the primary eigenvector of $\alpha W_{XY}(I-(1-\alpha)W_{YY})^{-1}W_{YX}$. Since the iterative process is a power method to calculate primary eigenvectors, the ranking score will finally get converge.

For authority ranking, the time complexity is $O(t|E|)$, where t is the iteration number and $|E|$ is the number of links in the graph. Notice that, $|E| = O(d|V|) \ll |V|^2$ in a sparse network, where $|V|$ is the number of total objects in the network and d is the average link per each object.

Different from simple ranking, authority ranking gives importance measure to each object according to the whole network, rather than the immediate neighborhoods, by the score propagation over the whole network.

4.3 Alternative Ranking Functions

Although in this section, we only illustrate two possible ranking functions, the general ranking functions are not confined to these two types. Also, in reality, ranking function is

[2] For example, a statistician may want to change the rules referring to conferences to journals; whereas a bibliographic database that collects papers from all the bogus conferences may need even more sophisticated rules (extracted from the domain knowledge) to guard the ranking quality.

not only related to the link property of an information network, but also depended on the hidden ranking rules used by people in some specific domain. Ranking functions should be combined with link information and user rules in that domain. For example, in many other science fields, journals should be given higher weight when considering an author's rank. Finally, ranking function on heterogeneous networks with more types of objects can be similarly defined. For example, PopRank [13] is a possible framework to deal with heterogeneous network, which takes into account both the impact within the same type of objects and its relations with other types of objects. The popularity scores of objects are mutually reinforced through the relations with each other, with different impact factors of different types. When ranking objects in information networks, junk or spam entities are often ranked higher than deserved. For example, authority ranking can be spammed by some bogus conferences that accept any submit papers due to their huge publication number. Techniques that could best use expert knowledge such as TrustRank [7] could be used, which can semi-automatically separate reputable, good objects from spam ones, toward a robust ranking scheme.

5. THE RANKCLUS ALGORITHM

In this section, we introduce RankClus algorithm based on bi-type network and ranking function defined in Section 4. Given the bi-type network $G = \langle\{X \cup Y\}, W\rangle$, suppose that we have a random partition on target type X already, how can we use the conditional ranks to improve the clustering results further? Intuitively, for each conference cluster, which could form a research area, the rank of authors conditional on this area should be very distinct, and quite different from the rank of authors in other areas. Therefore, for each cluster X_k, conditional rank of Y, $\vec{r}_{Y|X_k}$, can be viewed as a rank distribution of Y, which in fact is a measure for cluster X_k. Then, for each object x in X, the distribution of object y in Y can be viewed as a mixture model over K conditional ranks of Y, and thus can be represented as a K dimensional vector in the new measure space. We first build the mixture model and use EM algorithm to get the Component coefficients for each object in Section 5.1, then propose the distance measure between object and cluster in Section 5.2, then summarize the algorithm in Section 5.3, and finally give some discussions on extending RankClus to arbitrary information networks in Section 5.4.

5.1 Mixture Model of Conditional Rank Distribution

Example 5.1 (Conditional Rank as Cluster Feature) Conditional ranks on different clusters are very different from each other, especially when these clusters are correctly partitioned. Still using the data of the two-research-area example proposed in Section 1, we rank two hundred authors based on two conference clusters, and the two conditional rank distributions are shown in Figure 1. From the figure, we can clearly see that DB/DM authors rank high relative to DB/DM conferences, while rank extremely low relative to HW/CA conferences. The situation is similar for HW/CA authors. ■

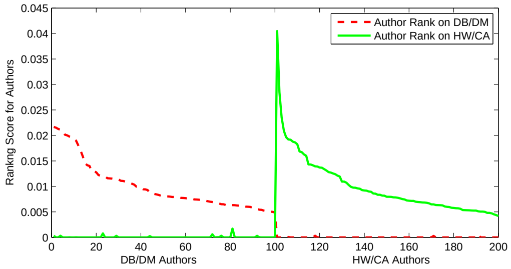

Figure 1: Authors' Rank Distribution on Different Clusters

From Example 5.1, one can see that conditional rank distributions for attribute type on each cluster are quite different from each other, and can be used as measures to characterize each cluster. This gives us the intuition to model the distribution for each object x in X over Y as a mixture distribution of K conditional rank distributions over Y. Here, we only consider the simple case that there are no links between target objects, *i.e.*, $W_{XX} = 0$, and more complex situations will be discussed in Section 5.4.

5.1.1 *Mixture Model for Each Target Object*

Suppose we now know the clustering results for type X, which are $X_1, X_2, \ldots,$ and X_K. Also, according to some given ranking function, we have got conditional rank distribution over Y on each cluster X_k, which is $\vec{r}_{Y|X_k}(k = 1, 2, \ldots, K)$, and conditional rank over X, which is $\vec{r}_{X|X_k}(k = 1, 2, \ldots, K)$. For simplicity, we use $p_k(Y)$ to denote $\vec{r}_{Y|X_k}$ and $p_k(X)$ to denote $\vec{r}_{X|X_k}$ in the following deduction. For each object $x_i (i = 1, 2, \ldots, m)$ in X, it follows a distribution $p_{x_i}(Y) = p(Y|x_i)$ to generate a link between x_i and y in Y. Moreover, this distribution could be considered as a mixture model over K component distributions, which are attribute type's conditional rank distributions on K clusters. We use $\pi_{i,k}$ to denote x_i's coefficient for component k, which in fact is the posterior probability that x_i from cluster k. Thus, $p_{x_i}(Y)$ can be modeled as:

$$p_{x_i}(Y) = \sum_{k=1}^{K} \pi_{i,k} p_k(Y), \text{ and } \sum_{k=1}^{K} \pi_{i,k} = 1. \tag{6}$$

$\pi_{i,k}$ in fact is the probability that object x_i belonging to cluster k, $p(k|x_i)$. Since $p(k|x_i) \propto p(x_i|k)p(k)$, and we have already known $p(x_i|k)$, which is the conditional rank of x_i in cluster k. The goal is thus to estimate the prior of $p(k)$, which is the probability that a link between object x and y belongs to cluster k. In DBLP scenario, a link is a paper, and papers with the same conference and author will be considered as the same papers (since we do not have additional information to discriminate them). The cluster of conference, *e.g.*, DB conferences, can induce a subnetwork of conferences and authors with the semantic meaning of DB research area. $p(k)$ is the proportion of papers that belonging to the research area induced by the kth conference cluster. Notice that, we can just set the priors as uniform distribution, and then $p(k|x_i) \propto p(x_i|k)$, which means the higher its conditional rank on a cluster, the higher possibility that the object will belong to that cluster. Since conditional rank of X is the propagation score of conditional rank of Y, we can see that highly ranked attribute object has more

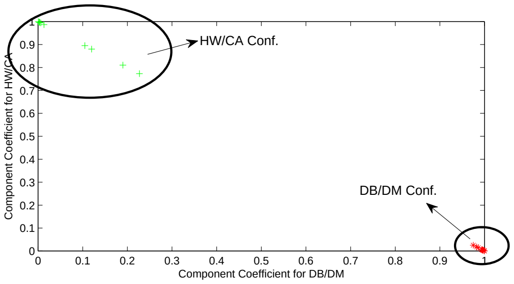

Figure 2: Conferences' Scatter Plot based on Two Component Coefficients

impact on determining the cluster label of target object.

To evaluate the model, we also make an independence assumption that an attribute object y_j issuing a link is independent to a target object x_i accepting this link, which is $p_k(x_i, y_j) = p_k(x_i)p_k(y_j)$. This assumption says once a author writes a paper, he is more likely to submit it to a highly ranked conference to improve his rank; while for conferences, they are more likely to accept papers coming from highly ranked authors to improve its rank as well.

Example 5.2 (Component Coefficients as Object Attributes) Following Ex. 5.1, each conference x_i is decomposed as a two dimensional vector $(\pi_{i,1}, \pi_{i,2})$, each dimension stands for the component coefficient. Figure 2 is the scatter plot for each conference's two component coefficients, and different shapes of points represent different areas the conferences really belong to. From the figure, we can see that DB/DM conferences and HW/CA conferences are separated clearly under the new attributes. ■

5.1.2 Parameter Estimation Using EM Algorithm

Next, let's address the problem to estimate the component coefficients in the mixture model. Let Θ be the parameter matrix, which is a $m \times K$ matrix: $\Theta_{m\times K} = \{\pi_{i,k}\}(i = 1, 2, \ldots, m; k = 1, 2, \ldots, K)$. Our task now is to evaluate the best Θ, given the links we observed in the network. For all the links W_{XY} and W_{YY}, we have the likelihood of generating all the links under parameter Θ as:

$$
\begin{aligned}
&L'(\Theta|W_{XY}, W_{YY}) = p(W_{XY}|\Theta)p(W_{YY}|\Theta)\\
=&\prod_{i=1}^{m}\prod_{j=1}^{n} p(x_i, y_j|\Theta)^{W_{XY}(i,j)} \prod_{j=1}^{n}\prod_{j=1}^{n} p(y_i, y_j|\Theta)^{W_{YY}(i,j)}
\end{aligned}
$$

where, $p(x_i, y_j|\Theta)$ is the probability to generate link $\langle x_i, y_j\rangle$, given current parameter. Since $p(W_{YY}|\Theta)$ does not contain variables from Θ, we only need to consider maximizing the first part of the likelihood to get the best estimation of Θ. Let $L(\Theta|W_{XY})$ be the first part of likelihood. As it is difficult to maximize L directly, we apply EM algorithm [1] to solve the problem.

In E-Step, we introduce hidden variable $z \in \{1, 2, \ldots, K\}$ for each link, which indicates the cluster label that a link $\langle x, y\rangle$ is from. The complete log likelihood thus can be written as:

$$
\begin{aligned}
&\log L(\theta|W_{XY}, Z)\\
=&\log \prod_{i=1}^{m}\prod_{j=1}^{n}(p_(x_i, y_j, z)|\Theta)^{W_{XY}(i,j)}\\
=&\log \prod_{i=1}^{m}\prod_{j=1}^{n}[p(x_i, y_j|z, \Theta)p(z|\Theta)]^{W_{XY}(i,j)}\\
=&\sum_{i=1}^{m}\sum_{j=1}^{n} W_{XY}(i,j)\log(p_z(x_i, y_j)p(z|\Theta))
\end{aligned}
$$

where, $p_z(x_i, y_j)$ is the probability to generate a link $\langle x_i, y_j\rangle$ from cluster z. By considering conditional rank of x_i and y_j as the probability that they will be visited in the network and assuming the independence between variables x and y, $p_z(x_i, y_j) = p_z(x_i)p_z(y_j)$,

Given the initial parameter is Θ^0, which could be set as $\pi^0_{i,k} = \frac{1}{K}$, for all i and k, expectation of log likelihood under current distribution of Z is:

$$
\begin{aligned}
&Q(\Theta, \Theta^0) = E_{f(Z|W_{XY}, \Theta^0)}(\log L(\theta|W_{XY}, Z))\\
=&\sum_{k=1}^{K}\sum_{i=1}^{m}\sum_{j=1}^{n} W_{XY}(i,j)\log(p_{z=k}(x_i, y_j)p(z=k|\Theta))p(z=k|x_i, y_j, \Theta^0)\\
=&\sum_{k=1}^{K}\sum_{i=1}^{m}\sum_{j=1}^{n} W_{XY}(i,j)\log(p_k(x_i, y_j)p(z=k|\Theta))p(z=k|x_i, y_j, \Theta^0)\\
=&\sum_{i=1}^{m}\sum_{k=1}^{K}\sum_{j=1}^{n} W_{XY}(i,j)\log(p(z=k|\Theta))p(z=k|x_i, y_j, \Theta^0)+\\
&\sum_{k=1}^{K}\sum_{i=1}^{m}\sum_{j=1}^{n} W_{XY}(i,j)\log(p_k(x_i, y_j))p(z=k|x_i, y_j, \Theta^0)
\end{aligned}
$$

For conditional distribution $p(z = k|y_j, x_i, \Theta^0)$, it can be calculated using Bayesian rule as follows,

$$
\begin{aligned}
&p(z=k|y_j, x_i, \Theta^0)\\
\propto& p(x_i, y_j|z=k, \Theta^0)p(z=k|\Theta^0)\\
\propto& p^0_k(x_i)p^0_k(y_j)p^0(z=k)
\end{aligned}
\tag{7}
$$

In M-Step, in order to get the estimation for $p(z = k)$, we need to maximize $Q(\Theta, \Theta^0)$. Introducing Lagrange multiplier λ, we get:

$$
\frac{\partial}{\partial p(z=k)}[Q(\Theta, \Theta^0) + \lambda(\sum_{k=1}^{K} p(z=k) - 1)] = 0
$$

$$
\Rightarrow \sum_{i=1}^{m}\sum_{j=1}^{n} W_{XY}(i,j)\frac{1}{p(z=k)}p(z=k|x_i, y_j, \Theta^0) + \lambda = 0
$$

Thus, integrating with Eq. (7), we can get the new estimation for $p(z = k)$ given previous Θ^0:

$$
p(z=k) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} W_{XY}(i,j)p(z=k|x_i, y_j, \Theta^0)}{\sum_{i=1}^{m}\sum_{j=1}^{n} W_{XY}(i,j)}. \tag{8}
$$

Finally, each parameter $\pi_{i,k}$ in Θ is calculated using Bayesian rule:

$$
\pi_{i,k} = p(z=k|x_i) = \frac{p_k(x_i)p(z=k)}{\sum_{l=1}^{K} p_l(x_i)p(z=l)} \tag{9}
$$

By setting $\Theta^0 = \Theta$, the whole process can be repeated. At each iteration, updating rules from Eqs. (7)-(9) are applied, and finally Θ will converge to a local maximum.

5.2 Cluster Centers and Distance Measure

After we get the estimations for component efficient for each target object x_i by evaluating mixture models, x_i can be represented as a K dimensional vector $\vec{s}_{x_i} = (\pi_{i,1}, \pi_{i,2}, \ldots, \pi_{i,K})$. The centers for each cluster can thus be calculated accordingly, which is the mean of $\vec{s}_{x_i}$ for all x_i in each cluster:

$$\vec{s}_{X_k} = \frac{\sum_{x \in X_k} \vec{s}(x)}{|X_k|}$$

where $|X_k|$ is the size of the cluster k.

Next, the distance between an object and cluster $D(x, X_k)$ is defined by 1 minus cosine similarity:

$$D(x, X_k) = 1 - \frac{\sum_{l=1}^{K} \vec{s}_x(l)\vec{s}_{X_k}(l)}{\sqrt{\sum_{l=1}^{K}(\vec{s}_x(l))^2}\sqrt{\sum_{l=1}^{K}(\vec{s}_{X_k}(l))^2}}. \quad (10)$$

An alternative method is to use component coefficient $p_{i,k}$ as the similarity measure of object x_i and cluster k directly. However, through both our analysis and experiment results, we found that it is not a wise choice. When initial clusters are randomly partitioned, the initial conditional ranking would be quite similar to each other. In this case, it's possible that all the objects are mixed together and all belong to one cluster in terms of $p_{i,k}$. An example is shown in Figure 3(b), conditional rank distributions on Cluster 1 and Cluster 2 is similar to each other, and rank distribution on Cluster 2 is dominating Cluster 1 in more data points. As a result, almost every object will have a higher coefficient relative to Cluster 2. If we simply assign the object according to this coefficient, no object will be assigned to Cluster 1. However, our definition of cluster center and distance measure can correctly assign each object to the correct cluster after several iterations. Our measure doesn't totally dependent on the clusters, especially when the cluster quality is not good, it could be a disaster to completely rely on component coefficients. However, we also consider the similarity between objects under the new measure space, even at first the measure feature is not that good, the similarity between them can still somehow be retained.

5.3 RankClus: Algorithm Summarization

The general idea of RankClus is first to convert each object into $\vec{s}_x$ based on the mixture model of current clustering, and then adjust objects into the nearest cluster X_k under the new attributes. The process repeats until clusters do not change significantly. During the process, clusters will be improved because similar objects under new attributes will be grouped together; ranking will be improved along with the better clusters, and thus offers better attributes for further clustering. In this section, we describe the algorithm in detail.

RankClus is mainly composed of three steps, put in an iterative refinement manner. First, rank for each cluster. Second, estimate the parameter Θ in the mixture model, get new representations $\vec{s}_x$ for each target object and $\vec{s}_{X_k}$ for each target cluster. Third, adjust each object in type X, calculate the distance from it to each cluster center and assign it to the nearest cluster.

The input of RankClus is bi-type information network $G = \langle\{X \cup Y\}, W\rangle$, the ranking function f, and the cluster number K. The output is K clusters of X with within-cluster rank scores for each x, and conditional rank scores for ean y. The algorithm works as follows, which is summarized in Table 4.

- Step 0: Initialization.
  In the initialization step, generate initial clusters for target objects, *i.e.*, assign each target object with a cluster label from 1 to K randomly.
- Step 1: Ranking for each cluster.
  Based on current clusters, calculate conditional rank for type Y and X and within-cluster rank for type X. In this step, we also need to judge whether any cluster is empty, which may be caused by the improper initialization or biased running results of the algorithm. When some cluster is empty, the algorithm needs to restart in order to generate K clusters.
- Step 2: Estimation of the mixture model component coefficients.
  Estimate the parameter Θ in the mixture model, get new representations for each target object and centers for each target cluster: $\vec{s}_x$ and $\vec{s}_{X_k}$. In practice, the iteration number t for calculating Θ only needs to be set to a small number. Empirically, $t = 5$ can already achieve best results.
- Step 3: Cluster adjustment.
  Calculate the distance from each object to each cluster center using Eq. (10) and assign it to the nearest cluster.
- Repeat Steps 1, 2 and 3 until clusters changes only by a very small ratio ε or the iteration number is bigger than a predefined number *iterNum*. In practice, we can set $\varepsilon = 0$, and $iterNum = 20$. Through our experiments, the algorithm will converge less than 5 rounds in most cases for the synthetic data set and around 10 rounds for DBLP data.

Example 5.3 (Mutual Improvement of Clustering and Ranking) We now apply our algorithm to the two-research-area example. The conditional rank and component coefficients for each conference at each iteration of the running procedure are illustrated in Figure 3 through (a)-(h). To better explain how our algorithm can work, we set an extremely bad initial clustering as the initial state. In Cluster 1, there are 14 conferences, half from DB/DM area and half from HW/CA area. Accordingly, Cluster 2 contains the remaining 6 conferences, which are ICDT, CIKM, PKDD, ASPLOS, ISLPED and CODES. We can see that the partition is quite unbalanced according to the size, and quite mixed according to the area. During the first iteration, the conditional rank distribution for two clusters are very similar to each other (Fig. 3(a)), and conferences are mixed up and biased to Cluster 2 (Fig. 3(b)), however we can still adjust their cluster label according to the cluster centers and most HW/CA conferences become into the Cluster 2 and most DB/DM conferences become Cluster 1. At the second iteration, conditional ranking is improved a little (shown in Fig.

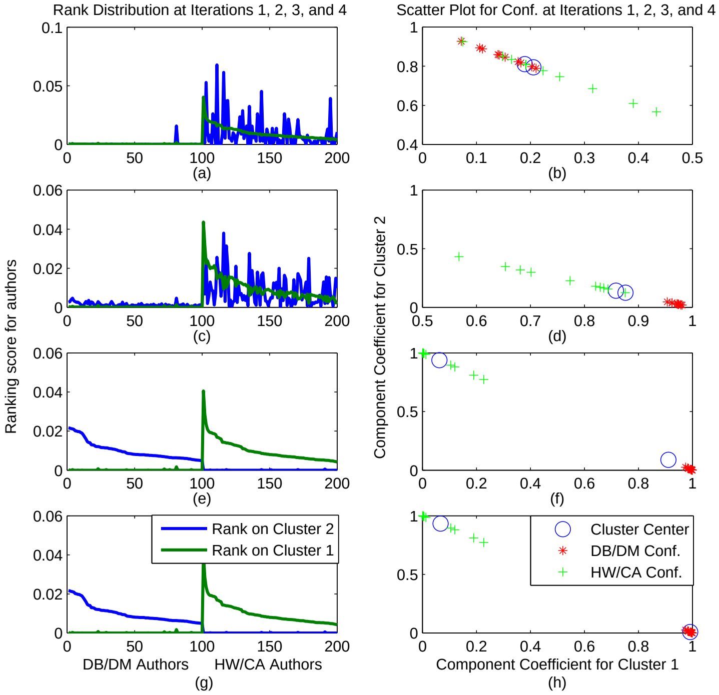

Figure 3: Mutual Improvement of Clusters and Ranking through Iterations

3(c)) since the clustering (Fig. 3(b)) is enhanced, and this time clustering results (Fig. 3(d)) are enhance dramatically, although they are still biased to one cluster (Cluster 1). At the third iteration, ranking results are improved dramatically. Clusters and ranks are further adjusted afterwards, both of which are minor refinements. ■

Table 4: RankClus Algorithm

```
Procedure: RankClus()
  Input:   Bi-type Information Network $G = \langle X, Y; W \rangle$,
           Ranking function $f$,
           Cluster Number $K$.
  Output:  $K$ clusters $X_i$, $\vec{r}_{X_i|X_i}$, $\vec{r}_{Y|X_i}$.
//Step 0: Initialization
1      $t = 0$;
2      $\{X_i^{(t)}\}_{i=1}^{K}$ = get initial partitions for $X$;
//Repeat Steps 1-3 until $< \varepsilon$ change or too many iterations
3      For (iter = 0; iter < iterNum && epsi > $\varepsilon$; iter++)
         //Step 1: Ranking for each cluster
4          if any of the clusters are empty, restart, goto Line 1;
5          For i = 1 to K
6              $G_i^{(t)}$ = get subgraph from $G$, using $X_i^{(t)}$, $Y$;
7              $(\vec{r}^{(t)}_{X_i|X_i}, \vec{r}^{(t)}_{Y|X_i}) = f(G_i^{(t)})$; $\vec{r}^{(t)}_{X|X_i} = W_{XY}\vec{r}^{(t)}_{Y|X_i}$;
8          End for
       //Step 2: Get new attributes for objects and cluster
9          Evaluate $\Theta$ for mixture model, thus get $\vec{s}_{x_i}$ for each
       object $x_i$;
10         For i = 1 to K
11             $\vec{s}^{(t)}_{X_k}$ = get centers for cluster $X_k^{(t)}$;
12         End for
       //Step 3: Adjust each object
13         For each object x in X
14             For i = 1 to K
15                 Calculate Distance $D(x, X_k^{(t)})$
16             End for
17             Assign x to $X_{k_0}^{t+1}$, $k_0 = \arg\min_k D(x, X_k^{(t)})$
18         End for
18     End For
```

At each iteration, the time complexity of RankClus is comprised of three parts: ranking part, mixture model estimation part and clustering adjustment part. For clustering adjustment, we need to compute the distance between each object (m) and each cluster (K), and the dimension of each object is K, so the time complexity for this part is $O(mK^2)$. For ranking, if we use simple ranking, the time complexity is $O(|E|)$. If we use authority ranking, the time complexity is $O(t_1|E|)$, where $|E|$ is the number of links, and t_1 is the iteration number of ranking. For mixture model estimation, at each round, we need to calculate $O(K|E| + K + mK)$ parameters. So, overall, the time complexity is $O(t(t_1|E| + t_2(K|E| + K + mK) + mK^2))$, where t is the iteration number of the whole algorithm and t_2 is the iteration number of the mixture model. If the network is a sparse network, the time is almost linear with the number of objects.

5.4 Discussion: Extensions to Arbitrary multi-typed Information Network

In the previous sections, the reasoning of RankClus is based on bi-type networks, with the constraint that there are no links between target objects (*i.e.*, $W_{XX} = 0$). However, RankClus can be applied to other information network as well. In this section, we introduce the basic idea to use RankClus in an arbitrary network: The key is to generate a new set of attributes *from every attribute type* for each object, and then RankClus algorithm proposed in Section 5.3 can be used directly.

1. **One-type information network.** For one-type information network $G = \langle\{X\}, W\rangle$, the problem can be transformed into bi-type network settings $G = \langle\{X \cup Y\}, W\rangle$, where $Y = X$.

2. **Bi-type information network with $W_{XX} \neq 0$.** For bi-type information network that $W_{XX} \neq 0$, the network can be transformed into a three-type network $G = \langle\{X \cup Z \cup Y\}, W\rangle$, where $Z = X$. In this situation, two sets of parameters Θ_Z and Θ_Y can be evaluated separately, by considering links of W_{XZ} and W_{XY} independently. Therefore, for each object x, there should be $2K$ parameters. The first K parameters are its mixture model coefficients over conditional rank distributions of X, while the second K parameters are its mixture model coefficients over conditional rank distributions of Y.

3. **Multi-typed information network.** For multi-typed information network $G = \langle\{X \cup Y_1 \cup Y_2 \cup \ldots \cup Y_N\}, W\rangle$, the problem can be solved similarly to the second case. In this case, we need to evaluated N sets of parameters, by considering conditional ranks from N types: $Y_1, Y_2, \ldots, Y_N$. So, each object can be represented as a NK dimensional vector.

6. EXPERIMENTS

In this section, we will show the effectiveness and efficiency of RankClus algorithm, based on both synthetic and real datasets.

6.1 Synthetic Data

In order to compare accuracy among different clustering algorithms, we generate synthetic bi-type information networks, which follow the properties of real information networks similar to DBLP. Configuration parameters for generating synthetic networks with different properties are as follows:

- Cluster number: K.
- Size of object sets and link distributions. In each cluster, set two types of objects: Type X and Type Y. The number of objects in X and Y are respectively N_x and N_y. The link distribution for each object follows Zipf's law with parameter s_x and s_y for each type. Zipf's law is defined by $f(k; s, N) = \frac{1/k^s}{\sum_{i=1}^{N} 1/i^s}$, which denotes the link frequency of an object that ranks in the k^{th} position.
- Transition probability matrix T, which denotes the probability that a link is generated from any two clusters.
- Link numbers for each cluster: P, which denotes the total number of links in each cluster.

In our experiments, we first fixed the scale of the network and the distribution of links, but change T and P to generate 5 kinds of networks with different properties, where T determines how much the clusters are separated and P determines the density of each cluster. We set $K = 3$, $N_x = [10, 20, 15]$, $N_y = [500, 800, 700]$, $s_x = 1.01$, and $s_y = 0.95$ for all the 5 configurations. Five different pairs of T and P are set as:

- Data1: medium separated and medium density.
  $P = [1000, 1500, 2000]$,
  $T = [0.8, 0.05, 0.15; 0.1, 0.8, 0.1; 0.1, 0.05, 0.85]$
- Data2: medium separated and low density.
  $P = [800, 1300, 1200]$,
  $T = [0.8, 0.05, 0.15; 0.1, 0.8, 0.1; 0.1, 0.05, 0.85]$
- Data3: medium separated and high density.
  $P = [2000, 3000, 4000]$,
  $T = [0.8, 0.05, 0.15; 0.1, 0.8, 0.1; 0.1, 0.05, 0.85]$
- Data4: highly separated and medium density.
  $P = [1000, 1500, 2000]$,
  $T = [0.9, 0.05, 0.05; 0.05, 0.9, 0.05; 0.1, 0.05, 0.85]$
- Data5: poorly separated and medium density.
  $P = [1000, 1500, 2000]$,
  $T = [0.7, 0.15, 0.15; 0.15, 0.7, 0.15; 0.15, 0.15, 0.7]$

In order to evaluate the accuracy of the clustering results, we adopt Normalized Mutual Information measure. For N objects, set cluster number as K, and two clustering results, let $n(i, j), i, j = 1, 2, \ldots, K$, the number of objects that has the cluster label i in the first cluster and cluster label j in the second cluster. From $n(i, j)$, we can define joint distribution $p(i, j) = \frac{n(i,j)}{N}$, row distribution $p_1(j) = \sum_{i=1}^{K} p(i, j)$ and column distribution $p_2(i) = \sum_{j=1}^{K} p(i, j)$. NMI is defined as follows:

$$NMI = \frac{\sum_{i=1}^{K} \sum_{j=1}^{K} p(i, j) \log(\frac{p(i,j)}{p_1(j)p_2(i)})}{\sqrt{\sum_{j=1}^{K} p_1(j) \log p_1(j) \sum_{i=1}^{K} p_2(i) \log p_2(i)}}$$

We compared RankClus implemented with two ranking functions, which are Simple Ranking and Authority Ranking, with state-of-the-art spectral clustering algorithm, which is the k-way Ncut algorithm proposed in [15], implemented with two similarity matrix generation methods, which are Jaccard Coefficient and SimRank [9]. Results for accuracy is in Figure 4. For each network configuration, we generate 10 different datasets and run each algorithm 100 times. From the results, we can see that, two versions of RankClus outperform in the first 4 data sets. RankClus with Authority ranking function is even better, since authority ranking gives a better rank distribution, as it is able to utilize the information of the whole network. Through the experiments, we observe that performance of two versions of RankClus and the NCut algorithm based on Jaccard coefficient are highly dependent on the data quality, in terms of cluster sperateness and link density. SimRank has a very stable performance. Further experiments show that the performance of SimRank will deteriorate when the data quality is rather poor (when average link for each target object is 40, the NMI accuracy becomes as low as 0.6846).

In order to check the scalability of each algorithm, we set four different size networks, in which both the object size and link size are increasing by a factor of 2. The average time used by each algorithm for each dataset is summarized in Figure 5. We can see that compared with the time-consuming SimRank algorithm, RankClus is also very efficient and scalable.

Impact of iteration number in the mixture model on clustering accuracy is examined. Through Figure 6, we can see

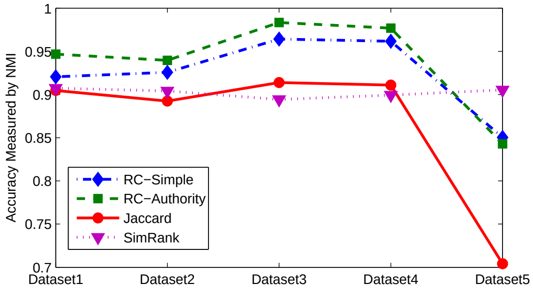

Figure 4: Accuracy of Clustering

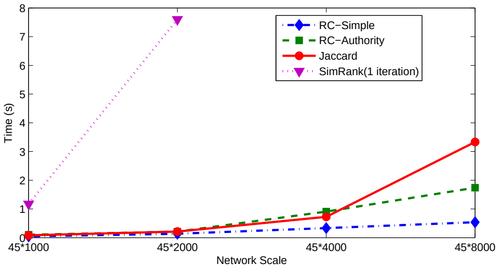

Figure 5: Efficiency Analysis

that when the iteration number is getting larger, the accuracy will first be improved then stable. In fact, even when the iteration number is set to a very small number, the results are still very good.

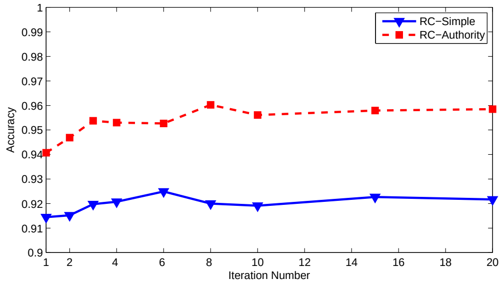

Figure 6: Impact of Iteration Number in Mixture Model

6.2 Real Data: The DBLP Data Set

We use the DBLP dataset to generate a bi-type information network for all the 2676 conferences and 20,000 authors with most publications, from the time period of year 1998 to year 2007. Both conference-author relationships and co-author relationships are used. We set cluster number $K = 15$, and apply RankClus with authority function proposed in Section 4.2, with $\alpha = 0.95$. We then pick 5 clusters, and show top-10 conferences from each cluster according to within-cluster scores. For clarifying, we also add research area labels manually to each cluster. The results are shown in Table 5.

Please note that the clustering and ranking of conferences and authors shown in Tables 5 and **??** have not used any keyword nor citation information, the information popularly used in most bibliographic data clustering or ranking systems. It is well recognized that citation information is crucial at judging the influence and impact of a conference or an author in a field. However, by exploring the publication entries only in the DBLP data, the RankClus algorithm can achieve comparable performance as citation studies for clustering and ranking conferences and authors. This implies that the collection of publication entries without referring to the keyword and citation information can still tell a lot about the status of conferences and authors in a scientific field.

7. DISCUSSION

In RankClus we assume that a user will specify a target type to be the clustering type. Although according to the view of algorithm we can specify any type as target type, some type of objects would be clustered better in terms of semantic meaning, quality of clustering, and the efficiency. In the DBLP data set, conference is a better choice for clustering, since it has less number of distinct values, which means a smaller number of clusters can summarize the whole network well; also, it has the better semantic meaning of research area than authors. Moreover, considering computational issues, we find that the convergence speed of RankClus would be much much lower when using author as target type.

Efficiency of RankClus could be further improved if we wisely select the starting value. First, the quality of initial clusters determines the number of iteration of the algorithm. We may use some seed objects to form initial clusters to start the RankClus processing. Second, the initial value of the rank score is also very important to the convergence speed. When we do Authority Ranking, Simple Ranking score could be a good starting point. Another way to improve efficiency is to first filtering the globally lowly ranked attribute objects, which could reduce the scale of network. Since the lowly ranked attribute objects only have low impact to determine the cluster label of target objects.

RankClus is the first piece of work that utilizes ranking as cluster feature to improve clustering results and tightly integrates ranking and clustering. However, there are many other issues need to be considered in the future.

First, currently we have only performed experiments on the bi-type information network. It is still not that clear on how we can utilize additional information and constraints in the RankClus process, such as how to add citation information and text information to the bibliographic data and how we can utilize the additional information to make refined clustering and ranking. This will be an interesting topic for further study.

Second, the empirical rules and its associated weight computation formulas proposed in this study may not be directly transferable to other problem domains. When applying the RankClus methodology to other bibliographic data, such as PubMed, we need to re-consider the empirical rules for ranking functions. When applying the methodology to non-bibliographic data sets, both new ranking functions and the

Table 5: Top-10 Conferences in 5 Clusters Using RankClus

	DB	Network	AI	Theory	IR
1	VLDB	INFOCOM	AAMAS	SODA	SIGIR
2	ICDE	SIGMETRICS	IJCAI	STOC	ACM Multimedia
3	SIGMOD	ICNP	AAAI	FOCS	CIKM
4	KDD	SIGCOMM	Agents	ICALP	TREC
5	ICDM	MOBICOM	AAAI/IAAI	CCC	JCDL
6	EDBT	ICDCS	ECAI	SPAA	CLEF
7	DASFAA	NETWORKING	RoboCup	PODC	WWW
8	PODS	MobiHoc	IAT	CRYPTO	ECDL
9	SSDBM	ISCC	ICMAS	APPROX-RANDOM	ECIR
10	SDM	SenSys	CP	EUROCRYPT	CIVR

semantics of links need to be reconsidered.

Third, the quality of ranking function is important to the accuracy of clustering, as it can capture the distinct feature for clusters. However, as we can see, ranking function is highly related to different domains, how we can automatically extract rules based on a small partial ranking results given by experts could be another interesting problem.

8. CONCLUSIONS

In this paper, we propose a novel clustering framework called RankClus to integrate clustering with ranking, which generates conditional ranking relative to clusters to improve ranking quality, and uses conditional ranking to generate new measure attributes to improve clustering. As a result, the quality of clustering and ranking are mutually enhanced, which means the clusters are getting more accurate and the ranking is getting more meaningful. Moreover, the clustering results with ranking can provide more informative views of data. Our experiment results show that RankClus can generate more accurate clusters and in a more efficient way than the state-of-the-art link-based clustering method. There are still many research issues to be explored in the RankClus framework. We have identified a few of them in Section 7. Clearly, more research is needed to further consolidate this interesting framework and explore its broad applications.

9. REFERENCES

[1] J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models, 1997.

[2] 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.

[3] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather: a cluster-based approach to browsing large document collections. pages 318–329, 1992.

[4] DBLP. The dblp computer science bibliography. http://www.informatik.uni-trier.de/ ley/db/.

[5] J. E. Gentle and W. HSrdle. *Handbook of Computational Statistics: Concepts and Methods*, chapter 7 Evaluation of Eigenvalues, pages 245–247. Springer, 1 edition, 2004.

[6] C. L. Giles. The future of citeseer. In *10th European Conference on PKDD (PKDD'06)*, page 2, 2006.

[7] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large data bases (VLDB'04)*, pages 576–587. VLDB Endowment, 2004.

[8] J. E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences*, 102:16569, 2005.

[9] G. Jeh and J. Widom. SimRank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD conference (KDD'02)*, pages 538–543. ACM, 2002.

[10] W. Jiang, J. Vaidya, Z. Balaporia, C. Clifton, and B. Banich. Knowledge discovery from transportation network data. In *Proceedings of the 21st ICDE Conference (ICDE'05)*, pages 1061–1072, 2005.

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

[12] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[13] Z. Nie, Y. Zhang, J.-R. Wen, and W.-Y. Ma. Object-level ranking: Bringing order to web objects. In *Proceedings of the fourteenth International World Wide Web Conference (WWW'05)*, pages 567–574. ACM, May 2005.

[14] S. Roy, T. Lane, and M. Werner-Washburne. Integrative construction and analysis of condition-specific biological networks. In *Proceedings of AAAI'07*, pages 1898–1899, 2007.

[15] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[16] A. Sidiropoulos, D. Katsaros, and Y. Manolopoulos. Generalized h-index for disclosing latent facts in citation networks. *CoRR*, abs/cs/0607066, 2006.

[17] X. Yin, J. Han, and P. S. Yu. Linkclus: Efficient clustering via heterogeneous semantic links. In *Proceedings of the 32nd VLDB conference (VLDB'06)*, pages 427–438, 2006.

[18] O. Zamir and O. Etzioni. Grouper: A dynamic clustering interface to web search results. pages 1361–1374, 1999.