

RCHIG: An Effective Clustering Algorithm with Ranking

Jianwen TAO

College of Information Engineering, Zhejiang Business Technology Institute

Ningbo City, China

Email: jianwentao@Yahoo.com.cn

Abstract—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 heterogeneous information graph. A novel clustering framework called RCHIG 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 RCHIG 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.

Index Terms—Binary Information Graph, Clustering, K -dimensional Vector, Link-based Clustering

I. 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 graphs. We call such interconnected graphs as heterogeneous information graphs, 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 graphs are ubiquitous and form a critical component of modern information infrastructure. Among them, heterogeneous graph is a special type of graph that contains objects of multiple types.

A great many analytical techniques have been proposed toward a better understanding of information graphs and their properties, among which are two prominent ones: ranking and clustering. On one hand, ranking evaluates objects of information graphs 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 graphs. 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 graphs, and hence be widely applied in different information graph settings.

Clustering and ranking are often regarded as orthogonal techniques, each of which is applied separately to information graph analysis. However, applying either of them over information graphs often leads to incomplete, or sometimes rather biased, analytical results. For instance, ranking objects over the global information graphs 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.

In this paper, we propose RCHIG, 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 graph. Our study shows that RCHIG 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 is successfully combined to analyze information graphs. 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 graphs.

2. We formally study how ranking and clustering can mutually reinforce each other in information graph analysis. A novel algorithm called RCHIG is proposed and its correctness and effectiveness are verified.

3. We perform a thorough experimental study on synthetic datasets in comparison with the state-of-the-art

algorithms, and the experimental results demonstrate the power of RCHIG.

II. PROBLEM DEFINITION

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

Definition 1. Binary Information Graph (BIG). Given two types of object sets X and Y , where $X = \{x_1, x_2, \dots, x_m\}$, and $Y = \{y_1, y_2, \dots, y_n\}$, graph $G = \langle V, E \rangle$ is called a binary type information graph 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 $W_{(m+n) \times (m+n)} = \{w_{o_i o_j}\}$ be the adjacency matrix of links, where $w_{o_i o_j}$ 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 binary type information graph. 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-graph 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 binary type graph $G = \langle \{X \cup Y\}, W \rangle$, if a function $f: G \rightarrow (\vec{r}_X, \vec{r}_Y)$ 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 graph G .

The aim of ranking in information graph 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 is a single-type information graph with web pages as its objects. For the binary type information graph defined in the DBLP data [4], we will provide two ranking functions.

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 binary type graphs, 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 graph, 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 considering 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-graph $G' = \langle \{X' \cup Y\}, W' \rangle$ 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-graph 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 graph 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-graph 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-graph 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 .

Based on these definitions, our goal of this paper can be summarized as follows: given a binary type graph $G = \langle \{X \cup Y\}, W \rangle$, 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 $\vec{r}_{Y|X_k}, k = 1, 2, \dots, K$.

III. 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 binary type graph similar to conference-author graph. In bibliographic graph, consider the binary type information graph 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 WXY as:

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

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 WY Y as:

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

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 graph, we define two ranking functions: Simple Ranking and Authority Ranking.

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

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

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 scores; even these papers are all in junk conferences. In fact, simple ranking evaluate importance of each object according to their immediate neighborhoods.

B. 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.

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,

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

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,

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

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:

$$\begin{cases} \vec{r}_X = \frac{W_{XY} \vec{r}_Y}{\|W_{XY} \vec{r}_Y\|} \\ \vec{r}_Y = \frac{W_{YX} \vec{r}_X}{\|W_{YX} \vec{r}_X\|} \end{cases}, \quad (4)$$

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 co-authors with many authors or many highly ranked authors.

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

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

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 graph, where $|V|$ is the number of total objects in the graph 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 graph, rather than the immediate neighborhoods, by the score propagation over the whole graph.

C. 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 not only related to the link property of an information graph, 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 graphs with more types of objects can be similarly defined. For example, PopRank [13] is a possible framework to deal with heterogeneous graph, 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 graphs, 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.

IV. THE RCHIG ALGORITHM DESIGNING

A. Mixture Model for Each Target Object

Suppose we now know the clustering results for type X, which are X_1, X_2, \dots, X_k . Also, according to some given ranking function, we have got conditional rank distribution over Y on each cluster X_k , which is $\vec{r}_{Y|X_k}$ ($k = 1, 2, \dots, K$), and conditional rank over X, which is $\vec{r}_{X|X_k}$ ($k = 1, 2, \dots, K$). 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 ($i = 1, 2, \dots, m$) 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:

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

$\pi_{i,k}$ 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 sub graph 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 k^{th} 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 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.

B. 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: $\Theta_{m \times K} = \{\pi_{i,k}\} (i=1, 2, \dots, m; k=1, 2, \dots, K)$: Our task now is to evaluate the best Θ , given the links we observed in the graph. For all the links W_{XY} and W_{YY} , we have the likelihood of generating all the links under parameter Θ as:

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

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] solve the problem.

In E-Step, we introduce hidden variable $z \in \{1, 2, \dots, K\}$ 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:

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

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 graph 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 $\pi_{i,k}^0 = 1/K$, for all i and k , expectation of log likelihood under current distribution of Z is:

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

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

$$p(z = k | y_j, x_i, \Theta^0) \\ \propto p(y_j, x_i | z = k, \Theta^0) p(z = k | \Theta^0), \tag{7} \\ \propto p_k(x_i) p_k^0(y_j) p^0(z = k)$$

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.

C. 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 $\vec{s}_{xi} = (\pi_{i,1}, \pi_{i,2}, \dots, \pi_{i,K})$. The centers for each cluster can thus be calculated accordingly, which is the mean of \vec{s}_{xi} for all x_i in each cluster:

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

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:

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

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 $p_{i,k}$. 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, and the similarity between them can still somehow be retained.

D. RCHIG Algorithm Summarization

The general idea of RCHIG 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.

RCHIG 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}_{Xk} 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 RCHIG is binary type information graph $G = \langle \{X \cup Y\}, W \rangle$, 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 each y . The algorithm works as follows.

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}_{Xk} . 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.

Step 4: 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 $\epsilon = 0$, 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[4] data.

At each iteration, the time complexity of RCHIG 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 graph is a sparse graph, the time is almost linear with the number of objects.

E. Discussion

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

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

2. Binary type information graph with $W_{XX} \neq 0$. For binary type information graph that $W_{XX} \neq 0$, the graph can be transformed into a three-type graph $G = \langle \{X \cup Z \cup Y\}, W \rangle$, 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 graph. For multi-typed information graph $G = \langle \{X \cup Y_1 \cup Y_2 \dots \cup Y_N\}, W \rangle$, the problem can be solved similarly to the second case. In this case, we need to evaluate N sets of parameters, by considering conditional ranks from N types: Y_1, Y_2, \dots, Y_N . So, each object can be represented as a NK dimensional vector.

V. EXPERIMENTS AND RESULTS

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

1. Cluster number: K.
2. 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 $f(k, s, N) = (1/k^s) / \sum_{i=1}^N 1/i^s$, which denotes the link frequency of an object that ranks in the k^{th} position.
3. Transition probability matrix T, which denotes the probability that a link is generated from any two clusters.
4. Link numbers for each cluster: P, which denotes the

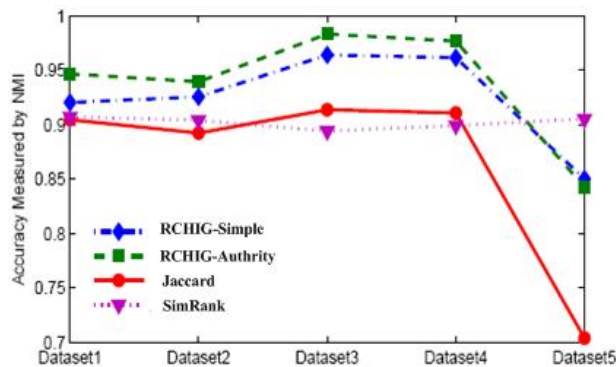


Figure 1. Accuracy of Clustering.

total number of links in each cluster.

In our experiments, we first fixed the scale of the graph and the distribution of links, but change T and P to generate 5 kinds of graphs with different properties, where T determines how many 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:

1. Data1: medium separated and medium density.
 $P = [1000, 1500, 2000]$,
 $T = [0.8, 0.05, 0.15, 0.1, 0.8, 0.1, 0.1, 0.05, 0.85]$
2. Data2: medium separated and low density.
 $P = [800, 1300, 1200]$,
 $T = [0.8, 0.05, 0.15, 0.1, 0.8, 0.1, 0.1, 0.05, 0.85]$

3. Data3: medium separated and high density.
 $P = [2000, 3000, 4000]$,
 $T = [0.8, 0.05, 0.15, 0.1, 0.8, 0.1, 0.1, 0.05, 0.85]$
4. Data4: highly separated and medium density.
 $P = [1000, 1500, 2000]$,
 $T = [0.9, 0.05, 0.05, 0.05, 0.9, 0.05, 0.1, 0.05, 0.85]$
5. Data5: poorly separated and medium density.
 $P = [1000, 1500, 2000]$,
 $T = [0.7, 0.15, 0.15, 0.15, 0.7, 0.15, 0.15, 0.15, 0.7]$

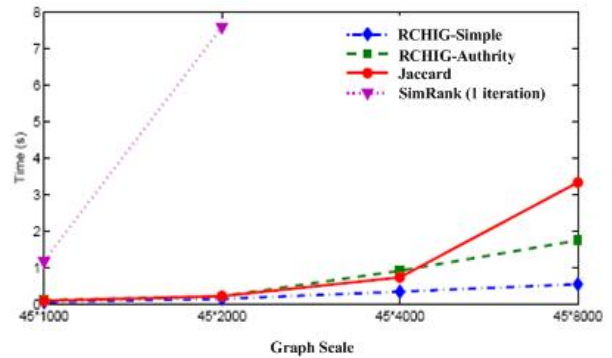


Figure 2. Efficiency Analysis.

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 $n(i, j)$, $i, j = 1, 2, \dots, K$, 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 $p(i, j) = n(i, j)/N$, 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 RCHIG 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 N-cut algorithm proposed in [15], implemented with two similarity matrix generation methods, which are Jaccard Coefficient and SimRank [9]. Results for accuracy are in Figure 1. For each graph configuration, we generate 10 different datasets and run each algorithm 100 times. From the results, we can see that, two versions of RCHIG outperform in the first 4 data sets. RCHIG 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 graph. Through the experiments, we observe that performance of two versions of RCHIG and the N-Cut algorithm based on Jaccard coefficient are highly dependent on the data quality, in terms of cluster separateness 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

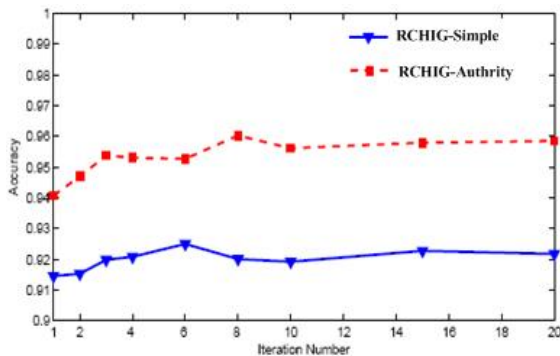


Figure3. Impact of Iteration Number in Mixture Model.

NMI accuracy becomes as low as 0.6846).

In order to check the scalability of each algorithm, we set four different size graphs, 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 2. We can see that compared with the time-consuming SimRank algorithm, RCHIG is also very efficient and scalable. Impact of iteration number in the mixture model on clustering accuracy is examined. Through Figure 3, we can see 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.

VI. CONCLUSIONS

In this paper, we propose a novel clustering framework called RCHIG 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 RCHIG 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 RCHIG framework. Clearly, more research is needed to further consolidate this interesting framework and explore its broad applications.

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. GyÅongyi, 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-Washburn. 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.

Jianwen Tao was born in Wuhan, Hubei Province, P.R.China, in April 15, 1973. He received B. Sc degree and M. Sc degree in computer application from Hubei Polytechnic University, P.R. China in 1995 and 1999 respectively. He is currently a teacher in the Institute of Information engineer, Zhejiang Business technology Institute, Zhejiang, P.R.China.

His research interest includes data mining, pattern recognition and parallel processing. He has published more than 20 papers in journals and conferences.