Integrating Clustering and Ranking on Hybrid Heterogeneous Information Network

Ran Wang[1], Chuan Shi[1], Philip S. Yu[2,3], and Bin Wu[1]

[1] Beijing University of Posts and Telecommunications, Beijing, China
{wangran51,shichuan,wubin}@bupt.edu.cn
[2] University of Illinois at Chicago, IL, USA
[3] King Abdulaziz University Jeddah, Saudi Arabia
psyu@cs.uic.edu

Abstract. Recently, ranking-based clustering on heterogeneous information network has emerged, which shows its advantages on the mutual promotion of clustering and ranking. However, these algorithms are restricted to information network only containing heterogeneous relations. In many applications, networked data are more complex and they can be represented as a hybrid network which simultaneously includes heterogeneous and homogeneous relations. It is more promising to promote clustering and ranking performance by combining the heterogeneous and homogeneous relations. This paper studied the ranking-based clustering on this kind of hybrid network and proposed the ComClus algorithm. ComClus applies star schema with self loop to organize the hybrid network and uses a probability model to represent the generative probability of objects. Experiments show that ComClus can achieve more accurate clustering results and do more reasonable ranking with quick and steady convergence.

Keywords: Clustering, Ranking, Heterogeneous Information Network, Probability Model.

1 Introduction

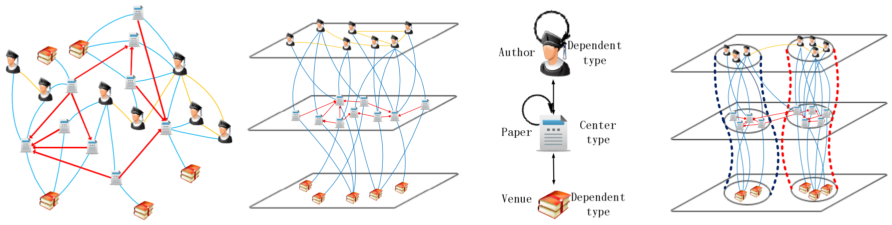
Information network analysis is an increasingly important direction in data mining in the past decade. Many analytical techniques have been developed to explore structures and properties of information networks, among which clustering and ranking are two primary tasks. The clustering task [1] partitions objects into different groups with similar objects gathered and dissimilar objects separated. Spectral method [1,4] is widely used in graph clustering. The ranking task [6,10,12] evaluates the importance of objects based on some ranking function, such as PageRank [12] or MultiRank [10]. Clustering and ranking are often regarded as two independent tasks and they are applied separately to information network analysis. However, integrating clustering and ranking makes more sense in many applications [2-3,11]. On one hand, the knowledge of important objects in a cluster helps to understand this cluster; on the other hand, knowing clusters is benefited to make more elaborate ranking. Some preliminary works have explored this issue [11].

J. Pei et al. (Eds.): PAKDD 2013, Part I, LNAI 7818, pp. 583–594, 2013.

Although it is a promising way to do clustering and ranking together, previous approaches confine it to a “pure” heterogeneous information network which does not consider the homogeneous relations among same-typed objects. For example, RankClus [2] only considers relations between two-typed objects; NetClus [3] just considers relations among center type and attribute types. However, in many applications, the networked data are more complex. They include heterogeneous relations among different-typed objects as well as homogeneous relations among same-typed objects. Taking bibliographic data as an example which is shown in Fig. 1(a), papers, venues, authors and their relations construct a heterogeneous information network. Simultaneously, the network also includes the citation relations among papers and the social network among authors. It is important to cluster on such a hybrid network which includes heterogeneous and homogeneous relations at the same time. The hybrid network can more authentically represent real networked data. Moreover, more information from heterogeneous and homogeneous relations is promising to promote the performance of clustering and ranking.

Although it is important to integrate clustering and ranking on the hybrid network, it is seldom studied due to the following challenges. 1) It is difficult to effectively organize networked data. The hybrid network is more complex than either of them. The way to organize the network not only needs to effectively represent objects and their relations but also benefits for clustering and ranking analysis. 2) It is not easy to integrate information from heterogeneous and homogeneous relations to improve clustering and ranking performances. It is obvious that more information from different sources can help to obtain better performances. However, we need to design an effective mechanism to make full use of information from these two networks.

In this paper, we study the ranking based clustering problem on a hybrid network and propose a novel ComClus algorithm to solve it. A star schema with self loop is applied to organize the hybrid network. The ComClus employs a probability model to represent the generative probability of objects and the experts model and generative method are used to effectively combine the information from heterogeneous and homogeneous relations. Moreover, through applying the probability information of objects, we propose ComRank to identify the importance of objects based on ComClus. Experiments on DBLP show that ComClus achieves better clustering and ranking accuracy compared to well-established algorithms. In addition, ComClus has better stability and quicker convergence.



(a) Bibliographic data (b)Hybrid network (c)Star schema with self loop (d)Clusters on hybrid network

Fig. 1. An example of clustering on bibliographic data

2 Problem Formulation

In this section, we give the problem definition and some important concepts used in this paper.

Definition 1. ***Information Network.*** Given $K+1$ types of nodes, V^k is a vertex set, denoted by $V^k = \{v_0^k, v_1^k, v_2^k, \ldots\ldots, v_n^k\}$, where v_n^k represents the $n\text{-}th$ node belonging to the $k\text{-}th$ type. An information network can be represented as a weighted network $G = < V, E, W >$, if $V = \bigcup_{k=0}^{K} V^k$. E is a binary relation on V, and W is a weight mapping from an edge $e \in E$ to a real number $w \in R^+$. If $K \geq 2$ the information network G is ***heterogeneous information network;*** *and* ***homogeneous information networ*k** when $K = 1$.

For a network with multiple types of nodes, *K*-partite network [7,9] and star schema [3] are widely used. These network structures only have heterogeneous relations among different-typed nodes, without considering the homogeneous relations among same-typed nodes. However, real networked data are more complex hybrid networks where links exist not only in heterogeneous nodes but also in homogeneous nodes. So we propose the star schema with self loop for this kind of networks.

Definition 2. ***Star schema with self loop network.*** An information network $G = < V, E, W >$ on *K+1* types of nodes $V = \bigcup_{k=0}^{K} V^k$ is called star schema with self loop network, $E = E_{homo} \cup E_{hete}$ and $E_{homo} \cap E_{hete} = \emptyset$. If $\forall e = < v_i^0, v_j^k > \in E_{hete}$, $v_i^0 \in V^0 \wedge v_j^k \in V^k (k \neq 0)$. If $\forall e = < v_i^k, v_j^k > \in E_{homo}$, $v_i^0 \in V^0 \wedge v_j^0 \in V^0$ ($k = 0$) or $v_i^k \in V^k \wedge v_j^k \in V^k (k \neq 0)$. Type $V^0$is called the center type (denoted as V^c), and $V^k (k \neq 0)$ is called dependent types (denoted as V^d).

E_{homo} is the links set among the same-typed nodes (called homo-link) and E_{hete} is the links set among the different-typed nodes (called hete-link). Then the hete-link can be written as e<v_i^c, v_j^d>, representing the link between center node and dependent node. The homo-link is the link between two same-typed nodes, which is denoted as e<v_i^c,v_j^c> or e<v_i^d,v_j^d>.

Fig. 1 shows such an example. For a complex bibliographical data (see Fig. 1(a)), we can organize it as a hybrid network which includes heterogeneous network among different layers and homogeneous network on the same layer in Fig.1 (b). As shown in Fig. 1(c), the hybrid network can be represented with a star schema with self loop where “paper” is the center type, while “venue” and “author” are dependent types.

Now, we can formulate the problem of clustering on hybrid network. Given a network $G = < V, E, W >$, $V = \bigcup_{k=0}^{K} V^k$ and the cluster number N, our goal is to find a clusters set $C = \bigcup_{n=1}^{N} C_n$, where C_n is defined as $C_n = < G', P_n >$. $G'$is a subnet of G, $E' \subseteq E, V' \subseteq V$ and $\forall e = < v_i^p, v_j^q > \in E'$. The probability function P_n represents the possibility that node v_i^p belongs to clusterC_n, $P_n(v_i^p) \in [0,1]$, and $\sum_{n=1}^{N} P_n(v_i^p) = 1$. In our solution, we restrict probability function of center node $P_n(v_i^p) \in \{0,1\}$, and for dependent node v_j^d, P_n is the successive probability measure from 0 to 1.

3 The ComClus Algorithm

After introducing the basic framework of ComClus, this section describes the ComClus in detail and then proposes ComRank for estimating the importance of objects.

3.1 The Framework of ComClus

The basic idea of ComClus is to determine the memberships of center nodes and then estimate the memberships of dependent nodes by center nodes. We consider that the probability of center node is estimated by two probabilities: homogeneous probability and heterogeneous probability. The homogeneous probability of center node depends on its homo-links. The heterogeneous probability of center node is generated by the dependent nodes that are correlated with it. In order to co-consider the heterogeneous and homogeneous probability for center nodes, generative method and experts model are used to mix these two types information. Finally, we estimate the posterior probability for center node according to the Bayesian rule and reassign the memberships of center nodes. The ComClus will iteratively calculate posterior probability until the memberships do not change. Algorithm 1 shows the basic framework of ComClus.

Algorithm 1. ComClus: Detecting *N* clusters on hybrid information network

```
Input: Cluster number N and hybrid network G
Output: Membership of center node, the posterior probability of dependent node
Begin:
    Randomly partition on network G
    Calculate global probability of center node for smoothing: $p(v_i^c|G)$
    repeat
        foreach subnet $G_n \subseteq G$
            Calculate the homogeneous probability of center node: $p( v_i^c|C, G_n)$
            Calculate the conditional probability of dependent node: $p(v_i^d|G_n)$
            Calculate the heterogeneous probability of center node: $p(v_i^c|D^i, G_n)$
            Calculate the mixed probability: $p(v_i^c|G_n)$
        end
        Calculate the center node posterior probability: $p(G_n|v_i^c)$ and Reassign
    until $\vec{D}(V_i^c)$ convergence obtained
    Calculate the dependent node posterior probability: $p(G_n|v_i^d)$
End
```

3.2 Homogeneous Probability for Center Node

The homogeneous probability of v_i^c depends on its homo-links and denotes as $p(v_i^c|C, G)$. $p(v_i^c|C, G)$ represents the fraction of links that the center node v_i^c connects to other center nodes on G. This idea is inspired by a general phenomenon that a node has higher probability to connect with nodes within the same cluster.

For convenience, $hodeg(\ v_i^c|G)$ denotes the number of homo-links of v_i^c and the number of in-degree of center node v_i^c on homogeneous network is denoted as $in(v_i^c|G)$.

$$p(\ v_i^c|C,G) = \frac{hodeg(\ v_i^c|G)}{\sum_{i=1}^{|V^c|} hodeg(\ v_i^c|G)} \tag{1}$$

$$QuotedRate(\ v_i^c|G) = \frac{in(\ v_i^c|G)}{\sum_{i=1}^{|V^c|} in(\ v_i^c|G)} \tag{2}$$

The value of $QuotedRate(v_i^c|G)$ is calculated by the quoted times of v_i^c on G, which will be used to rank (in Sect.3.7 Eq. (11)) and filter the unimportant nodes (in Sect.3.3 Eq. (3)) in our algorithm. The center node v_i^c has higher possibility to be assigned into a cluster with higher $p(\ v_i^c|C,G)$. Therefore, the clustering result will benefit from the homogeneous information.

3.3 Conditional Probability for Dependent Node

We consider that the heterogeneous probability of center node v_i^c is generated by its related dependent nodes v_i^d. Therefore, we need to estimate the probability of v_i^d, which can be represented as $p\left(v_i^d\middle|G\right) = p(d|G) \times p\left(v_i^d\middle|d\ ,G\right)$. The probability of dependent type d being selected is $p(d\ |G) = \frac{|V^d|}{|V|}$, where $|V^d|$ is the number of nodes in dependent type d layer, and $|V|$ is the number of all nodes in G. After the type d being selected, the probability $p\left(v_i^d\middle|d\ ,G\right)$ can be estimated. We utilize the two dependent types d_a, d_b to mutually estimate the probability for $p\left(v_i^{d_a}\middle|d_a\ ,G\right)$ and $p\left(v_i^{d_b}\middle|d_b\ ,G\right)$. D^i is the related dependent type set of v_i^c. Take $p\left(v_i^{d_a}\middle|d_a\ ,G\right)$ as an instance. By taking advantage of the homogeneous information of $v_i^{d_a}$, we set $p\left(v_i^{d_a}\middle|d_a\ ,G\right) = \frac{hodeg(v_i^{d_a}|G)}{\sum_{i=1}^{|V^{d_a}|} hodeg(v_i^{d_a}|G)}$ at the beginning of iteration. We consider the center node v_i^c is the medium between $v_i^{d_a}$ and $v_i^{d_\mathrm{b}}$. Naturally, an important medium v_i^c should have a higher $QuotedRate(\ v_i^c|G)$ than an ordinary one. Besides, we use θ as a filter factor to expand the $QuotedRate(\ v_i^c|G)$ gap among ent v_i^c. Repeat calculating (4) and (5) until the convergence is obtained.

$$\theta = \begin{cases} 1\ if\ QuotedRate(\ v_i^c|G) < avgQuotedRate(V^c|G) \\ \theta\ if\ QuotedRate(\ v_i^c|G) \geq avgQuotedRate(V^c|G) \\ 0\ otherwise \end{cases} \tag{3}$$

$$score(\ v_i^c|G) = \ \theta \times QuotedRate(\ v_i^c|G) \times \sum_{i=1}^{|V^{d_b}|} \frac{e{<}v_i^c, v_i^{d_b}{>}{\times}p\left(v_i^{d_b}\middle|d_b\ ,G\right)}{hedeg(v_i^{d_b})} \tag{4}$$

$$p\left(v_i^{d_a}\middle|d_a\ ,G\right) = \sum_{i=1}^{|V^c|} \frac{e{<}v_i^c,v_i^{d_a}{>}{\times}score(\ v_i^c|G)}{hedeg(v_i^{d_\mathrm{a}})} \tag{5}$$

where $hedeg(v_i^{d_a})$ is the number of hete-links of $v_i^{d_a}$on G. We run the same process for $v_i^{d_b}$ to get the probability $p\left(v_i^{d_a}\middle|d_a, G\right)$. As a result, the "productive" dependent nodes and the "barren" nodes can be distinguished obviously. Normalization method can be used when necessary.

3.4 Heterogeneous Probability for Center Node

After conditional probability of dependent nodes being figured out, we can estimate the heterogeneous probability for v_i^c. Here, we make an independency assumption that the dependent nodes generate the heterogeneous probability of center node independently. Given dependent node probabilities which are related to v_i^c, the heterogeneous probability of center node v_i^c can be denoted as $p(\, v_i^c|D^i, G)$.

$$p(v_i^c|D^i, G) = \prod_d^{D^i} \prod_{i=1}^{|V^d|} p\left(v_i^d\middle|d\,, G\right) \tag{6}$$

3.5 Mixed Probability for Center Node

Until now, we obtain the homogeneous and heterogeneous probability of center node v_i^c. Next, the major difficulty in estimating the probability measure is how to jointly consider the homogeneous and heterogeneous distribution of center nodes. To mix the two distributions, we employ two methods: a generative method of center node and a mixture of experts model [5].

In the generative method, we consider the center node v_i^c is generated by two parts: the homogeneous and heterogeneous information of v_i^c. The former is $p(\, v_i^c|C, G)$ and the latter is $p(v_i^c|D^i, G)$. We can calculate the conditional probability on hybrid network G as follows:

$$p(v_i^c|G) = p(\, v_i^c|C, G) \times p(v_i^c|D^i, G) \tag{7}$$

In experts model, we regard the homogeneous and heterogeneous information of v_i^c as "homogeneous expert" and "heterogeneous expert". Then we can evaluate mixed probability of center node according to its own distribution. The mixture of experts model is denoted as follows:

$$p(v_i^c|G) = \sum_{m=1}^{M} \pi_m p_m(v_i^c|G) \tag{8}$$

where $M = 2$ represents the number of experts. If $m = 1,$ the homogeneous expert takes into effect: $p_1(v_i^c|G)$= $p(\, v_i^c|C, G)$. If $m = 2$, the heterogeneous expert is activated as: $p_2(v_i^c|G) = p(v_i^c|D^i, G)$. $\pi_m = \frac{p_m(v_i^c|G)p(E_m)}{\sum_{m=1}^{M} p_m(v_i^c|G)p(E_m)}$ can be seen as the weight of corresponding expert, and we adopt *Softmax* function to compute it. $p(E_m)$ is the weight of expert *m*, which is proportional to the number of heter-links or homo-links of v_i^c. For example, the weight of homogeneous expert of v_i^c is calculated by the following formula: $p(E_1) = \frac{hodeg(v_i^c|G)+1}{hodeg(v_i^c|G)+\sum_d^{|D^i|} hedeg(v_i^d|G)}$. Because we only have two experts, the $p(E_2)$ is simply set as 1- $p(E_1)$. Obviously, the weight is dynamic for each v_i^c.

Both methods can evaluate the conditional probability of center node, which can be applied to different scenarios. The generative method equally treats the homogeneous and heterogeneous information, because it simply products homogeneous and heterogeneous probability. Therefore, the generative method is suitable for the hybrid network with the same scales of homogeneous and heterogeneous relations. The mixture of experts model can dynamically adjust the weights of distributions (by π_m). As a result, the method is more suitable for the hybrid network of which the homogenous and heterogeneous parts have different size.

Besides, to avoid zero probabilities, we smooth the distribution by the following formula: $p(v_i^c|G_n) = \lambda p(v_i^c|G_n) + (1-\lambda)p(v_i^c|G)$, where λ is a smoothing parameter. G is the whole hybrid network and G_n is the *n*-th subnet.

3.6 Posterior Probability for Nodes

In the previous subsection, we get the conditional probability of center node v_i^c by mixing two distributions. Now, we need to calculate the posterior probability $p(G_n|v_i^c)$ for each v_i^c, and reassign the memberships for center nodes. The posterior probability of center node can be calculated by Bayesian rule: $p(G_n|v_i^c) \propto p(v_i^c|G_n) \times p(G_n)$, where $p(v_i^c|G_n)$ is the conditional probability in cluster G_n and $p(G_n)$ represents the cluster size. However, the size of cluster G_n is not fixed. For the purpose of getting the $p(G_n)$, the EM algorithm can be used to get the local optimum $p(G_n)$ by maximizing the log likelihood of center nodes in different areas.

$$log\ P = \sum_{i=1}^{|V^c|} \log[\sum_{n=1}^{N+1} p(v_i^c|G_n) \times p(G_n)] \tag{9}$$

where $|V^c|$ is the size of V^c, and *N+1* represents the global distribution on *G*. The target is to maximize $log\ P$ and two iterative steps can be set to optimize the value *P*. We set $p^0(G_n) = \frac{1}{N+1}$ before the first iteration. The following two steps run iteratively until the convergence is obtained. $p^t(G_n|v_i^c) \propto p(v_i^c|G_n) \times p(G_n)$; $p^{t+1}(G_n) = \sum_{i=1}^{|V|} \frac{p^t(G_n|v_i^c)}{|V|}$. Finally, we will have a N dimensional indicator vector $\vec{D}(v_i^c)$, which is made up of posterior probability of v_i^c. Then we can calculate the indicator of membership for each center node with *K*-means.

After the iterative process is finished, the posterior probability of dependent node $p(G_n|v_i^d)$ can be evaluated by the average posterior probability of center nodes connecting with v_i^d. The notation S^i is a set of center nodes connecting with v_i^d and $|S^i|$ is the size of set S^i.

$$p(G_n|v_i^d) = \sum_{i=1}^{|S^i|} \frac{p(G_n|v_i^c)}{|S^i|} \tag{10}$$

3.7 Ranking for Nodes

As an additional benefit for ComClus, the posterior probabilities of nodes can be used for ranking nodes. Once the cluster process is finished, we can further figure out the rank of nodes in their cluster. We proposed a function (called ComRank) to evaluate the importance of nodes.

$$Rank(v_i^c|G_n) = QuotedRate(v_i^c|G) \times p(G_n|v_i^c) \tag{11}$$

where $p(v_i^c|G_n)$ is the probability of center node v_i^c. Generally, the rank of center node is proportional to its $QuotedRate(v_i^c|G)$. It is natural in many applications. Taking bibliographic network as an example, the goodness of a paper is decided by the number of citations to a large extent. Another factor of rank function is the posterior probability, which can be seen as a cluster coefficient and represents the degree of membership in that cluster. The rank of dependent node v_i^d can be computed according to the rank of center nodes connecting with it.

$$Rank(v_i^d|G_n) = \sum_{i=1}^{|S^i|} Rank(v_i^c|G_n) \times p(G_n|v_i^d) \tag{12}$$

4 Experiment

In this section, we evaluate the effectiveness of our ComClus algorithm, and compare it with the state-of-the-art methods on two data sets.

4.1 Data Set

The DBLP is a dataset of bibliographic information in computer science domain. We use it to build a hybrid network with three-typed nodes: papers (center type), venues (dependent type) and authors (dependent type). Homo-links among authors form a co-author network, and homo-links among papers form a paper citation network. Hete-links are the writing relation between authors and papers and the publication relation between venues and papers. We extract venues from different areas according to the categories of China Computer Federation (http://www.ccf.org.cn). Moreover, CCF provides three levels for ranking venues: A, B, C. The class A is top venues, such as KDD in *data mining* (DM). The class B is some famous venues such as SDM, ICDM. The class C is admitted venues such as WAIM. In the experiments, we extract two different-scaled subsets of the DBLP which are called DBLP-L and DBLP-S.

The DBLP-S is a small size dataset and it includes three areas in computer domain: *database, data mining, and information retrieval.* There are 21venues (7 venues for each area, covering three levels), 25,020 papers and 10,907 authors in DBLP-S. Two or three venues for each level are picked out.

The DBLP-L is a large dataset. There are eight areas included, which are *computer network, information security, computer architecture, theory, software engineering & programming language, artificial intelligence& pattern recognition, computer graphics, data mining& information retrieval &database*. There are 280 venues

(35 venues for each area), 275,649 papers, and 238,673 authors. For each area, five venues are in A level and fifteen venues are selected in B or C level.

In these two datasets, venues are labeled with their research areas. Moreover, in DBLP-S, we randomly label 1031 papers and 1295 authors with three research areas, which are used to evaluate the clustering accuracy. All the results are based on 20 runnings, and average results are shown.

4.2 Clustering Accuracy Comparison Experiments

For accuracy evaluation, we apply our method to cluster on both DBLP-S and DBLP-L. We compare ComClus with the representative ranking-based clustering algorithm NetClus which can be applied in heterogeneous networks organized as star schema. The smoothing parameter λ is fixed at 0.7 in both two algorithms. The filter factor θ in ComClus is 3. The clustering accuracy of paper is the fraction of nodes identified correctly. For author and venue nodes, the accuracy is the posterior probability fraction of nodes identified correctly. Results are shown in Table 1. The two different mixture methods of ComClus both have higher accuracy than NetClus. The lower deviation of ComClus implies that ComClus is steadier than NetClus. The results show that, the additional homogeneous relation utilized by ComClus is helpful for improving its accuracy as well as stability. In addition, ComClus with experts model achieves better performance than ComClus with generative method. We think the reason is that experts model considers the weight of heterogeneous and homogeneous information. In the following experiments, we use ComClus with experts model as the standard version of ComClus.

Table 1. Clustering accuracy comparison for different-typed nodes

Accuracy	ComClus(experts method)		ComClus(generative method)		NetClus	
	Mean	Dev.	Mean	Dev.	Mean	Dev.
Paper(DBLP-S)	**0.774**	0.019	0.766	0.021	0.715	0.066
Venue(DBLP-S)	**0.855**	0.018	0.777	0.028	0.739	0.067
Author(DBLP-S)	**0.731**	0.018	0.680	0.016	0.697	0.052
Venue(DBLP-L)	**0.681**	0.041	0.648	0.046	0.579	0.084

Since the hybrid network includes homogeneous network, we compare ComClus with those clustering algorithms on homogeneous network, where a representative spectral clustering algorithm Normalize Cut [4] is employed. We design the similarity of two nodes (i,j) as: $S(i,j) = cos(V_i, V_j)$, where V_i is the adjacent vector of node i. The result is shown in Table 2, which clearly illustrates that ComClus is better than Normalized Cut. ComClus combines the information from homogeneous and heterogeneous relations. It makes ComClus outperform Normalized Cut which only uses homogeneous network information.

Table 2. Clustering accuracy comparison on homogeneous network

Accuracy	ComClus	Normalized Cut
Paper Accuracy	**0.787**	0.457

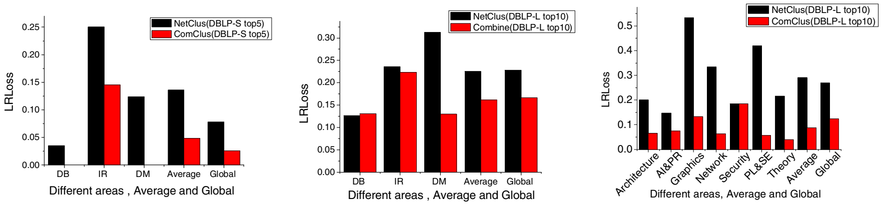
4.3 Ranking Accuracy Comparison Experiment

On DBLP-L, we make a ranking accuracy comparison between ComRank and AuthorithyRank which is a rank method in NetClus[3]. In this application, it is hard to definitely compare the goodness of two venues, whereas we can roughly distinguish their levels. For example, it is difficult to compare the ranking of SDM and ICDM. But we can safely say that SDM and ICDM are on the same level and they are worse than the top level venues (e.g., KDD) and better than the common level venues (e.g., WAIM). Inspired by *RankingLoss* measure [8], we define *LevelRankingLoss* to evaluate the disorder ratio of object pairs on their levels and it is abbreviated as *LRLoss*. Without loss of generality, we define *LRLoss* on bibliographic data. First, we define a triple to represent a venue:$RankTuple_i =< C_i, L_i, R_i >$, where C_i represents a venue, L_i is the level of C_i, $L_i \in \{\mathrm{A, B, C}\}$ (the recommended level of CCF). R_i is the rank number of C_i generated by the algorithms(the smaller, the better). The *LRLoss* is defined as follows.

$$LRLoss = \frac{1}{R}\sum_{i=1}^{R}\frac{|LossPair_i|}{|LossPair_i|+|\overline{LossPair_i}|} \tag{13}$$

where R is the size of Cartesian product of $RankTuple$ set and $LossPair_i = \{< RankTuple_i, RankTuple_j > | L_i < L_j, R_i > R_j \; or \; L_i > L_j, R_i < R_j\}$. Here, $\overline{LossPair_i}$ denotes the complementary set. $|LossPair_i|$ is the number of misordered pairs for $RankTuple_i$. For example, $RankTuple_1$= <KDD, A, 2>, $RankTuple_2$=<ICDM, B, 1> can be seen as one *LossPair* for $RankTuple_1$.

We select the top 5 and top 10 venues in different areas and then calculate *LRLoss* for them. Additionally, we also compare the accuracy of the global rank on both ComRank and NetClus. Results are shown in Fig2.



(a) 3 areas top5 venues on DBLP-S (b) 3 areas top10 venues on DBLP-L (c) 8 areas top 10 venues on DBLP-L

Fig. 2. Ranking accuracy comparison (The smaller *LRLoss*, the better)

The results clearly show that ComRank better ranks these venues, since its *LRLoss* is lower than that of AuthorityRank on all research areas. We think the additional homogeneous information utilized by ComRank contributes to its better ranking performance.

4.4 Case Study

In this section, we further show the performance of ComRank with a ranking case study.

Table 3. Top 15 venues with global rank on DBLP-S

ComRank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Venue	SIGMOD	VLDB	SIGIR	ICDE	KDD	PODS	WWW	CIKM	ICDM	EDBT	PKDD	WSDM	PAKDD	WebDB	DEXA
#Papers	2428	2444	2509	2832	1531	940	1501	2204	1436	747	680	198	1030	972	1731
Level	A	A	A	A	A	A	B	B	B	B	B	B	B	C	C
AuthorityRank	1	2	3	4	5	6	7	8	**9**	**10**	11	12	13	14	15
Venue	VLDB	ICDE	SIGMOD	SIGIR	KDD	WWW	CIKM	ICDM	**PODS**	**DEXA**	PAKDD	EDBT	PKDD	WSDM	ECIR
#Papers	2444	2832	2428	2509	1531	1510	2204	1436	**940**	**1731**	1030	747	680	198	575
Level	A	A	A	A	A	B	B	B	**A**	**C**	B	B	B	B	C

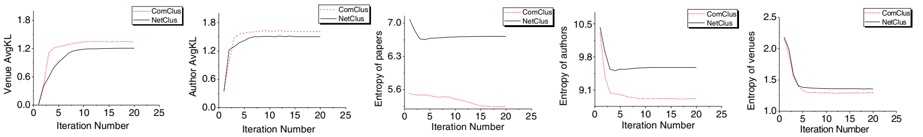
Table 3 shows the top 15 venues ranked by ComRank and AuthorityRank on DBLP-S. The results show that the ranks of venues generated by ComRank are all consistent with the recommended level by CCF. However, there are some disordered venues in AuthorityRank, which implies that AuthorityRank is sensitive to the number of papers. That is, AuthorityRank tends to rank a venue publishing many papers with a higher value. For example, AuthorityRank ranks PODS with a low value and DEXA with a relatively high value because PODS published not many papers and DEXA published so many papers. In contrast, ComRank considers the citation information from homogeneous network. So ComRank avoids these shortcomings.

4.5 Convergence and Stability Experiments

For observing the convergence, we compare each cluster probability distribution with global distribution by average KL divergence [3]. Next, we use entropy to measure the unpredictability of cluster and prove the algorithm stability.

$$AvgKL(V^d) = \frac{1}{N}\sum_{n=1}^{N} D_{KL}(p(v_i^d|G_n)||p(v_i^d|G)) \tag{14}$$

$$AvgEntropy(V^p) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{|V^p|} p\left(v_i^p\middle|G_n\right) \times log\, p\left(v_i^p\middle|G_n\right) \tag{15}$$



(a)*AvgKL* of venues (b) *AvgKL* of authors (c)*AvgEntropy* of papers (d)*AvgEntropy* of authors (e)*AvgEntropy* of venues

Fig. 3. The change of *AvgKL* and *AvgEntropy* of nodes with iteration number

As shown in Fig. 3(a) and (b), the convergence of our algorithm is faster than NetClus. From the results shown in Fig. 3(c), (d) and (e), we can observe that ComClus achieves lower $gEntropy$. The reason is that ComClus prevents the negative effects of unimportant paper by the factor θ. Besides, in ComRank, the distribution information of objects comes from heterogeneous and homogeneous relations. However, the distribution information of objects in NetClus is only from heterogeneous network. More information helps ComClus fast converge and achieve steady solution.

5 Conclusions

In this paper, we proposed a new ranking-based clustering algorithm ComClus on heterogeneous information networks. Different from conventional clustering methods, ComClus can group different-typed objects on a hybrid network which includes the homogeneous network and heterogeneous relations together. Through applying probability information in ComClus, ComClus can also rank the importance of objects. The experiments on real datasets have demonstrated that our algorithm can generate more accurate cluster and rank with quicker and steadier convergence.

Acknowledgments. It is supported by the National Natural Science Foundation of China (No. 60905025, 61074128, 71231002). This work is also supported by the National Basic Research Program of China (2013CB329603) and the Fundamental Research Funds for the Central Universities.

References

1. Shen, H., Cheng, X.: Spectral Methods for the Detection of Network Community Structure: a Comparative Analysis. J. Stat. Mech., P10020 (2010)
2. Sun, Y., Han, J., Zhao, P., Yin, Z., Cheng, H., Wu, T.: Rankclus: Integrating Clustering with Ranking for Heterogeneous Information Network Analysis. In: EDBT, pp. 565–576 (2009)
3. Sun, Y., Yu, Y., Han, J.: Ranking-based Clustering of Heterogeneous Information Networks with Star Network Schema. In: KDD, pp. 797–806 (2009)
4. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation. In: CVPR, pp. 731–737 (1997)
5. Jacobs, R.A., Jordan, M.I., Nowlan, S., Hinton, G.E.: Adaptive Mixtures of Local Experts. Neural Computation 3, 79–87 (1991)
6. Zhou, D., Orshanskiy, S., Zha, H., Giles, C.: Co-ranking Authors and Documents in a Heterogeneous Network. In: ICDM, pp. 739–744 (2007)
7. Liu, X., Murata, T.: Detecting Communities in K-partite K-uniform (Hyper) Networks. JCST 26(5), 778–791 (2011)
8. Zhang, M.L., Zhang, K.: Multi-label Learning by Exploiting Label Dependency. In: KDD, pp. 999–1008 (2010)
9. Long, B., Wu, X., Zhang, Z.M., Yu, P.S.: Unsupervised Learning on K-partite Graphs. In: KDD, pp. 317–326 (2006)
10. Michael, K.N., Li, X., Ye, Y.: MultiRank: Co-ranking for Objects and Relations in Multi-relational Data. In: KDD, pp. 1217–1225 (2011)
11. Ailon, N., Charikar, M., Newman, A.: Aggregating Inconsistent Information: Ranking and Clustering. J. ACM 55(5) (2008)
12. Brin, S., Page, L.: The Anatomy of a Large-scale Hyper Textual Web Search Engine. Comput. Netw. ISDN Syst. 30(1-7), 107–117 (1998)