Heterogeneous Information Network based Ranking and Clustering of Mobile Apps

Shuangling Bai
Zhejiang University
Hangzhou, China
bling@zju.edu.cn

Liang Chen
Zhejiang University
Hangzhou, China
cliang@zju.edu.cn

Jian Wu
Zhejiang University
Hangzhou, China
wujian2000@zju.edu.cn

ABSTRACT

With the development of mobile Internet network and smart phone, the study on mobile Apps data attracts more and more attention in recent years. Many data mining tasks have been exploited on mobile Apps data, among which clustering and ranking are two fundamental tasks. Most existing algorithms utilize only one or two types of Apps information, while in this paper we propose to cluster and rank mobile Apps based on a heterogeneous information network, which models related data as a network including different types of objects and relations. In order to make ranking and clustering mutually enhance each other, we introduce a ranking-based clustering algorithm and make it suitable for the mobile Apps scenario.To evaluate the performance of the proposed approach, we employ a real-world large-scale mobile Apps dataset, which contains more than 500,000 Apps. The experimental results demonstrate the effectiveness of the proposed approach.

Keywords

Mobile Apps, Clustering, Ranking, Heterogeneous Information Network

1. INTRODUCTION

In the past few years, with the rapid development of mobile Internet and smart phone, there is an explosive growth of the number of mobile Apps (Applications). App is constantly changing people's life. Today, Apps market is full of different kinds of Apps, which contains a large amount of data, that is, lots of information. Therefore, the study of Apps data is of great significance. On one hand, the comprehensive analysis of large Apps data can help us to understand users' behavior and provide better personalized service for users. For example, personalized Apps recommendation recommends more appropriate Apps to target users by exploring implicit structure of the Apps and the implied relationship between users. Apps recommendation can help improve the discovery experience for customers and support developers by helping drive adoption of their Apps and services. On the other hand, we can make use of these data to find more effective promotion platforms for advertising.

Current mobile application platforms are mainly divided into two types: Android platform and iOS platform. Up to March of 2014, there are more than 500,000 Apps hosted on 360 Mobile Phone Assistant [1]. The Apps downloads catalogs provided in this platform is based on the category or downloads ranking, that is to say, this App download platform does not provide personalized download directory, which means all users see the same list of Apps regardless of their tastes and preferences. Thus, if we can carry on the clustering analysis, users with similar taste will be clustered in the same group. Then we can provide more personalized service for these users. We can combine the result of clustering and the existing App recommendation technology to provide users the most optimized result. It can not only help to improve the efficiency but also can improve the recommendation accuracy.

In previous work, researchers mainly consider the similarity measure between Apps, pay attention to define similarity functions, and calculate the similarity based on browsing history or scoring record. Then using the similarity calculation results to explore the potential of the cluster. At present, the App information used for clustering analysis is so monotonous that it ignores a lot of useful information. In view of this, we can consider using all types of information of Apps to form a heterogeneous network. This heterogeneous network can contains objects of different types. After that, we can extract useful information to understand the potential structure in the heterogeneous information network. This step can be considered as the pre-processing for subsequent predictive modeling task.

Clustering is one of the important methods for understanding data and obtaining useful information. Organizing data into sensible groupings is one of the most fundamental modes for understanding and learning[3]. At present, there is no specialized clustering framework designed for Apps network. In this paper, we combine four types of Apps information to form a heterogeneous network, then use ranking and clustering simultaneously for analyzing. First of all, we process Apps data to build a heterogeneous information network then use a special ranking algorithm to get ranking results,

[1]http://zhushou.360.cn/

the ranking results can be used to estimate Apps posterior probability. In practice, we can obtain clustering distribution and ranking distribution of Apps and objects of other types. The results can be utilized to analyze Apps information network and understand information which is not explicitly stated. The main contributions of this paper are summarized as follows:

1. As far as we know, our work is the first to consider using more than one types of Apps information to do Apps clustering and advocate making use of both ranking and clustering simultaneously to analyze Apps information network.
2. We have a real large amounts of data from industry. The Apps number is more than 500,000 and each Apps can be assigned to one of 23 categories, in addition, the Apps dataset also contains more than 64,000 companies and a large amount of description information.

The rest of this paper is organized as follows: Section 2 reviews the the related work. In Section 3, we introduce the details of two important parts in ranking-based clustering algorithm: Ranking Algorithm and Probabilistic Generative Model, while Section 4 evaluates the performance of Ranking-Based Clustering Algorithm on Apps dataset. Section 5 concludes this paper.

2. RELATED WORK

In the previous work, a number of novel clustering algorithm have been proposed for heterogeneous network. In this section, we briefly introduce some related work. The presentation is mainly divided into two parts:

- ***Clustering***. Finding community structure in a very large network is of great significance, A. Clauset et al. present a hierarchical agglomeration algorithm for extracting community structure from large network, their method is proved faster than many competing algorithm [2]. A recent clustering method provides a fairly general multi-way clustering framework for relation graphs [1]. In this algorithm, entities are simultaneously clustered based not only on their intrinsic attribute values but also on the multiple relations between the entities. In addition, Y. Sun et al. design an appliable probabilistic clustering model for heterogeneous information networks with incomplete attributes across objects and different types of links [7] and B. Long et al. propose a spectral clustering-based methods for K-partite graphs [4]. Y. Sun et al. presents a semi-supervised clustering algorithm named PathSelClus, which integrate meta-path selection with user guidance to generate different cluster results [9].
- ***Ranking-Based Clustering***. Ranking-based clustering is first proposed in [8]. This novel algorithm directly generates clusters integrated with ranking and is proved to be an effective algorithm, which can generate very reasonable clustering and ranking results, however this method is designed only for bi-typed heterogeneous networks. Later, a suitable framework is proposed for star networks with types more than two [10]. Then a novel method named ComClus is proposed for Hybrid Heterogeneous information network, this Hybrid Heterogeneous network contains star network with self loop[11]. On the basis of previous research, recently a new suitable method named HeProjI is proposed for more general heterogeneous information network[6]. HeProjI projects a general heterogeneous network into a sub-networks sequence and design an information transfer mechanism to keep the consistency among sub-networks.

Inspired by the NetClus algorithm proposed by Sun[8], we view Apps clustering as a ranking clustering problem. On one hand, we use a special ranking algorithm to get ranking results for objects of all types. On the other hand, we can use the ranking results to estimate Apps posterior probabilities and calculate posterior probabilities for objects from other types at the same time.

3. RANKING-BASED CLUSTERING ALGORITHM

Before detail the key algorithm we introduce several related concepts and notations:

Definition 1. **Star Network.** Given one objects set of *center type* denoted by $\alpha = \{a_1, a_2, \cdots, a_n\}$ and m objects sets of *surrounding types* denoted by $\beta_1 = \{b_1, b_2, \cdots, b_{n_1}\}$, $\cdots$, $\beta_m = \{b_1, b_2, \cdots, b_{n_m}\}$. A graph $\varrho = \langle \nu, \varepsilon, \omega \rangle$, of which $\nu = \bigcup_{i=1}^{m} \beta_i \bigcup \alpha$, ε is the set of links only between $a\,(a \in \alpha)$ and $b\,(b \in \beta)$, ω is a weight set of links. We call such weighted graph Star Network.

Definition 2. **Ranking Distribution and Clustering Distribution .** Given a type of objects set, denoted by $\chi = \{x_1, x_2, \cdots, x_n\}$, $R(\chi)$ is the ranking distribution of χ, it is satisfies $R(\chi) = \bigcup_{i=1}^{n} R(x_i)$, $R(x_i) \geq 0$ and $\sum_{i=1}^{n} R(x_i) = 1$. $C(\chi)$ is the clustering distribution of χ. $C(\chi) = \{(p_1, p_2, \cdots p_K)_{x_1}, (p_1, p_2, \cdots, p_K)_{x_2}, \cdots, (p_1, p_2, \cdots p_K)_{x_n}\}$. Where $K = k+1$, k is the number of clusters, for every object, $\sum_{i=1}^{K} p_i = 1$.

Star Network is a special heterogeneous network. In our experiment, we extract four types of data from our dataset to form a Star Network, of which ***center type*** is *App* and ***surrounding types*** are *Category*, *Author* and *Description*. Given an information network, objects' ranking distribution reflect their importance within their own type. **Ranking Algorithm** and **Probabilistic Generative Model** are two important parts of ranking-based clustering algorithm.

In this paper we combine them to do ranking and clustering at the same time. There are two types of weight matrix: (1) If the weight matrix is between App and Author or between App and Category, $w_{x,y}$ has only two value: 1 or 0, of which 1 means there is an edge between x and y and 0 means there is no link between the two objects; (2) If the weight matrix is between App and Description words, $w_{x,y}$ can be any integer greater than or equal to zero.

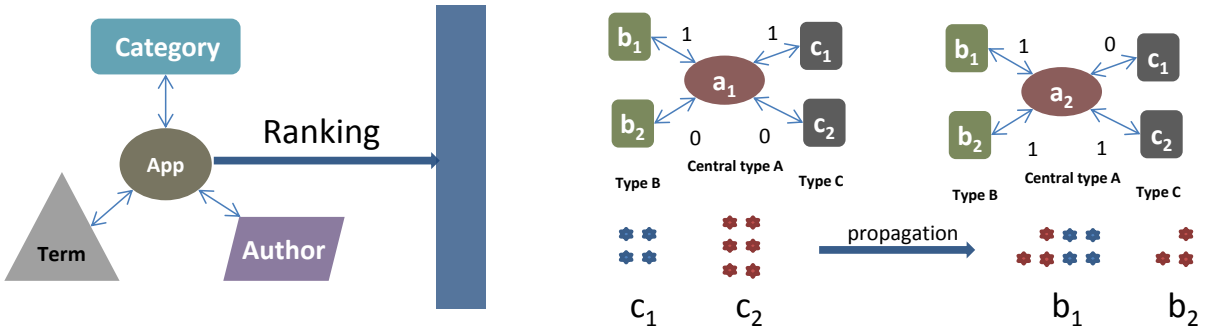

Figure 1: The Propagation of Ranking Values

3.1 The Ranking Algorithm

Given a ranking function we can get different ranking distributions for each type. An object's ranking value indicate the importance within its own type. For objects of the same type, their ranking values are very different in various clusters. Taking the special structure of the Apps information network into account, in our experiment we choose an effective propagation method as our main ranking algorithm. This method is similar to PageRank[5], but PageRank only applies to homogeneous network. The main idea of this ranking algorithm is that (1) an object given more authority means it has higher ranking score and (2) the authority can be propagated through network. The computational formula of authority is defined as follows:

$$P(x|\varrho) = \omega_{x\alpha}\omega_{\alpha y}P(y|\varrho) \tag{1}$$

where x and y are two surrounding types and α is the center type. $\omega_{x\alpha}$ is the weight matrix between type x and central type α. From this equation, we can see if two objects from type x and type y connect to a central object at the same time, the ranking score can be propagated through the central object. An object can get ranking values from objects of other types. A more detailed calculation is as follows:

$$\begin{aligned} \sigma_{a_x} &= \sum_{i=1}^{N_c} \frac{p(c_i|T_C, G)w_{c_i a_x}}{\sum_{j=1}^{N_a} w_{c_i a_j}} \\ p(b_y|T_B, G) &= \sum_{j=1}^{N_a} \frac{\sigma_{a_j} w_{b_y a_j}}{\sum_{i=0}^{N_b} w_{b_i a_j}} \end{aligned} \tag{2}$$

where b_y, c_i are objects of type B and type C and a_x is an object from center type A. N_c N_b and N_a is the objects number of type C, type B and type A. $w_{c_i a_j}$ is the weight of link between c_i and a_j. After calculation we can get ranking values of objects from type B. In order to explain the working mechanism of authority's propagation, we take an example to illustrate. Figure 1 shows connections of objects from three types, of which type B and type C are surrounding types and type A is a central type. We assume that the ranking values of object c_1 and c_2 are 0.4 and 0.6. After using ranking algorithm mentioned above, we can get ranking values of object b_1 and b_2, of which, c_2's authority propagate to b_1 and b_2 via a_2 and c_1's authority only propagate to b_1 via a_1. The ranking value of b_1 is 0.7 and the value of b_2 is 0.3. We can see, for b_1-related objects have more authority, b_1 get higher value. Considering the accuracy of results by iterative method can greatly enhanced, we determine the iterative ranking equation as:

$$\begin{aligned} P(\beta_1|\varrho) &= (\omega_{\beta_1\alpha}{\sigma^{-1}}_{\beta_1\alpha})(\omega_{\alpha\beta_2}{\sigma^{-1}}_{\alpha\beta_2})P(\beta_2|\varrho) \\ P(\beta_2|\varrho) &= (\omega_{\beta_2\alpha}{\sigma^{-1}}_{\beta_2\alpha})(\omega_{\alpha\beta_1}{\sigma^{-1}}_{\alpha\beta_1})P(\beta_1|\varrho) \end{aligned} \tag{3}$$

where β_1, β_2 are two types needed calculate ranking scores, α is a center type, $\omega_{\beta_1\alpha}$ is the weight matrix between β_1 and α, the same to $\omega_{\alpha\beta_2}$, $\omega_{\beta_2\alpha}$ and $\omega_{\alpha\beta_1}$. ${\sigma^{-1}}_{\beta_1\alpha}$, ${\sigma^{-1}}_{\alpha\beta_2}$, ${\sigma^{-1}}_{\beta_2\alpha}$ and ${\sigma^{-1}}_{\alpha\beta_1}$ are four diagonal matrices. The diagonal values of ${\sigma^{-1}}_{\beta_1\alpha}$, ${\sigma^{-1}}_{\alpha\beta_2}$, ${\sigma^{-1}}_{\beta_2\alpha}$ and ${\sigma^{-1}}_{\alpha\beta_1}$ are equal to column sum of $\omega_{\beta_1\alpha}$, $\omega_{\alpha\beta_2}$, $\omega_{\beta_2\alpha}$ and $\omega_{\alpha\beta_1}$. For a star network ϱ, the ranking distributions of β_1 and β_2 are calculated iteratively.

For Apps network, description type has little mutual information with Author type and Category type, therefore we use an simple ranking algorithm to get its ranking distribution, the equation is defined as follows:

$$p(a|\alpha, \varrho) = \frac{\sum_{b\in N_\varrho(a)} w_{ab}}{\sum_{a'\in\alpha}\sum_{b\in N_\varrho(a')} w_{a'b}} \tag{4}$$

where a is an object from type α and $N_\varrho(a)$ is the neighborhood of object a in network ϱ. In practice, we need a prior file to guide the first distribution of the objects.

3.2 Probabilistic Generative Model

The probabilistic generative model is designed for central objects of star network. The basic idea is that, for a central object, if the surrounding objects have high scoring in a sub network, it has a high probability to appear in this sub network. In other words, the surrounding objects together to generate the central object. At first we need to define the probability to visit an surrounding object b in a network ϱ:

$$p(b|\varrho) = p(\beta|\varrho) \times p(b|\beta, \varrho) \tag{5}$$

where $p(\beta|\varrho)$ is the probability to visit type β in network ϱ and $p(b|\beta, \varrho)$ is the probability that object b will be visited among all objects of type β in network ϱ. We assume that the probability to visit objects from different surrounding types and simultaneously visit two objects from the same type both independent. In Apps information network, App can be lack of describe word, this special situation may lead to zero probability problem. In order to avoid this, we join the global information before calculating posterior probabilities to smooth the results. Based on the above assumptions,

the probability to generate a central object a in sub-network ϱ_k is defined as follows:

$$p'(x|\beta,\varrho_k) = (1-\varsigma)p(x|\beta_x,\varrho_k) + \varsigma p(x|\beta_x,\varrho) \tag{6}$$

$$\begin{aligned} p(a|\varrho_k) &= \prod_{x\in N_{\varrho_k}(a)} p(x|\varrho_k)^{w_{a,x}} \\ &= \prod_{x\in N_{\varrho_k}(a)} p'(x|\beta_x,\varrho_k)^{w_{a,x}} p(\beta_x|\varrho_k)^{w_{a,x}} \end{aligned} \tag{7}$$

where $\omega_{a,x}$ is the weight of the link of edge and $k = \{1, 2, \cdots, K+1\}$, K is the number of clusters that we want to get. ς is a parameter to decide the degree of smooth. According to Bayesian rule, we can get the posterior probability of the central objects: $p(\varrho_k|a) \propto p(a|\varrho_k) \times p(\varrho_k)$. Where $p(\varrho_k|a)$ is the posterior probability of an central object a in sub-network ϱ_k. All posterior probabilities together to constitute the clustering distribution. We don't know the size of cluster k before clustering, therefore we can't directly calculate the posterior probability. In order to obtain an appropriate $p(\varrho_k)$ we consider to maximizes the likelihood of generating central objects in each sub-network and use EM algorithm to get the optimum for $p(\varrho_k)$:

$$logL = \sum_{a\in\alpha} log[\sum_{k=1}^{K+1} p(a|\varrho_k) \times p(\varrho_k)] \tag{8}$$

$$\begin{aligned} p^t(\varrho_k|a) &\propto p(a|\varrho_k)p^{(t)}(\varrho_k) \\ p^{t+1}(\varrho_k) &= \sum_{a\in\alpha} \frac{p^t(\varrho_k|a)}{|\alpha|} \end{aligned} \tag{9}$$

Where K is the cluster number decided by user, α is the central type and $|\alpha|$ is the objects number of type α. After we get posterior probability for central objects, we can calculate probability of each surrounding objects in different clusters via their neighbor relationship. The equation is defined as:

$$\begin{aligned} p(\varrho_k|b) &= \sum_{a\in N_{(b)}} p(\varrho_k, a|b) \\ &= \sum_{a\in N_{(b)}} \frac{p(\varrho_k|a)}{|N_{(b)}|} \end{aligned} \tag{10}$$

Where b is a surrounding object and $N_{(b)}$ is a central objects set which is connected to object b. For a surrounding object, its posterior probability in one cluster equals to the average values of its neighborhoods' probability belonging to the cluster. Following shows the detailed process of the proposed ranking-based clustering algorithm.

Algorithm 1 Framework of Ranking-based Clustering Algorithm

Input:
Cluster Number K,
Star Network $\varrho = \langle \nu, \varepsilon, \omega \rangle$
Output:
Clustering Distribution $C(\chi)$,
Ranking Distribution $R(\chi)$.

1: Randomly assign the central objects to different clusters to generate initial sub-networks ϱ_k, $k = 1, 2, \cdots, K$.
2: Get global rank of central type and surrounding types.
3: For $\varrho_k \in \varrho$:
(1)Generate the ranking probability of surrounding type $P(\beta_i|\varrho_k), i = 1, 2, \cdots, N_\beta$ and central type $P(\alpha|\varrho_k)$.
(2)Estimate the posterior probabilities of central objects:$P(\varrho_k|\alpha)$
(3)According to the posterior probabilities do reassignment of central types.
4: Repeat step 3 until the variation of last two clustering result in the allowed range.
5: Calculate probability for each surrounding objects in different clusters via their neighbor relationship $P(\varrho_k|\beta_i), i = 1, 2, \cdots, N_\beta$.

4. EXPERIMENT

In this section, we evaluate the ranking-based clustering algorithm on a large Apps dataset, including the ranking effectiveness and the clustering effectiveness.

4.1 Datasets

We have a large Apps dataset, which contains more than 50,000 Apps information. From this dataset we extract four types of information, including *App*, *Category*, *Description* and *Author*(company or individual). Relevant processing work is finished before our experiment. After removing those Apps which have incomplete information we use a words weighting technique named TF-IDF (term frequency-inverse document frequency) to extract the most characteristic describe words from App's description. Because the descriptive information include both Chinese sentences and English sentences, keywords extracting requires avoiding information loss as much as possible. After pre-processing, we picked out three groups Apps as our experimental data

Table 1: Detail Information of Three Datasets

Datasets	App	Category	Author	Key Words Number
Dataset1	303555	23	64301	≥0
Dataset2	301058	23	62475	≥5
Dataset3	293131	23	62475	≥10

Table 2: Category Information of Dataset3

Category	Apps Number	Sum Number	Cluster Category
Chess World	3022	72652	Games
Games	24		
Social Games	14		
RPG	3242		
Flight Shooting	5416		
Adventure	9460		
Online Games	1108		
Analog Auxiliary	1693		
Casual Puzzle	39466		
Sports Racing	4708		
Business Strategy	4504		
Map	44427	69460	Tools/Software
Office/Business	831		
Financial Planning	94		
Chat/Communication	6928		
Software	3		
Network/E-mail	1373		
System/Input	8604		
Video/Image	14513	65885	Video/Image
Wallpaper/Theme	51372		
Ebook	36613	85129	Reading/Learning
Reading/Learning	46970		
Children/Parenting	1546		

Table 3: The Top10 Companies in The Four Clusters

Ranking	G	Score	T/S	Score	V/I	Score	R/L	Score
1	Gallme	0.0222	Baoruan	0.0177	Moxiu	0.0510	Qidian	0.0954
2	Gameloft	0.0179	360	0.0080	Borui	0.0126	ReadingJoy	0.0928
3	iDreamSky	0.0172	BaiBanTec	0.0078	FuzhouHH	0.0059	Xxsy	0.0878
4	Pearlinpalm	0.0136	3G	0.0042	Xiaomi	0.0058	Readnovel	0.0829
5	C1wan	0.0122	FenghuaTec	0.0028	SitongTec	0.0055	HongXiu	0.0756
6	Yunyoyo	0.0114	NineWeiTec	0.0022	Guostudio	0.0020	iReader	0.0754
7	Meifeng	0.0113	Baidu	0.0019	Borui	0.0018	91Panda	0.0656
8	Tencent	0.0105	Tpadsz	0.0016	Palmstudio	0.0013	Tadu	0.0612
9	Ourpalm	0.0088	DadouSoft	0.0014	FuzhouTec	0.0011	Kanshu	0.0501
10	Mobage	0.0085	Wooboo	0.0012	JianyongLuo	0.0009	zhangyue	0.0367

which are named as Dataset1, Dataset2 and Dataset3. The detail information could be found in Table 1.

Category information is used to evaluate the accuracy of clustering. Table 2 shows the category information of Datase3. By further analyzing the data, we found that the number of different types of Apps has high disparities. It can be discovered from Table 2, in addition, the categories of Apps are quite ambiguous. For example, The App which labeled 'RPG' can also belong to 'Online Games' or 'Games'. The main reason is that the category labels are set unreasonably. For the simplicity of evaluation work, we choose four categories as target cluster categories, which contains Games(**G**), Tools/Software(**T/S**), Video/Image(**V/I**) and Reading/Learning(**R/L**). Before experiment we also need to choose one of the types and give it a prior distribution in various clusters. In practice, we choose *Description* type as Apps prior. These prior terms will be propagated in network at the beginning of procedure. In our experiment, priors for each cluster are around six or seven terms with different probabilities. These terms are related to different cluster, which can help attract objects for each cluster.

4.2 Evaluation Measures

Precision and Recall are two indicators frequently used in data mining, search engines and other related domains. For one cluster: (1) if the relevant objects are correctly clustered in this cluster we call them *true-positive* (tp) and the other relevant objects are called *false-negatives* (fn); (2) if the non-relevant objects are clustered in this cluster we call them *false-positive* (fp) and the other non-relevant objects are called *true-negatives*(tn). The Precision and Recall are respectively defined as:

$$\begin{aligned} Presicion &= \frac{tp}{tp+fp} \\ Recall &= \frac{tp}{tp+fn} \end{aligned} \tag{11}$$

As the Precision and Recall are sometimes contradictory, a more comprehensive indicator, F measure, is introduced for the evaluation. Actually, F measure is the the weighted harmonic mean of Precision and Recall, and the definition of F measure is showed in the following:

$$F = \frac{(a^2+1)precision * recall}{a^2(precision + recall)} \tag{12}$$

When parameter $a = 1$, we call it F_1 Measure. In our experiment we choose F_1 Measure as our overall indicator. F_1 combines the results of precision and recall, higher F1 means better experimental results.

4.3 Ranking Effectiveness Evaluate

Figure 2 illustrates the change of ranking value during the iteration process. Specifically, we respectively pick out 50 Apps from four clustering categories. From Fig. 2, it can be found that the ranking distributions for four clusters are quite overlapping(Fig.2(a)) after the first iteration. After several iterations, ranking result has a certain degree of improvement(Fig.2(b)). When the iterative procedure is finally completed, we can clearly see the significant improvement in ranking distribution(Fig.2(c)).

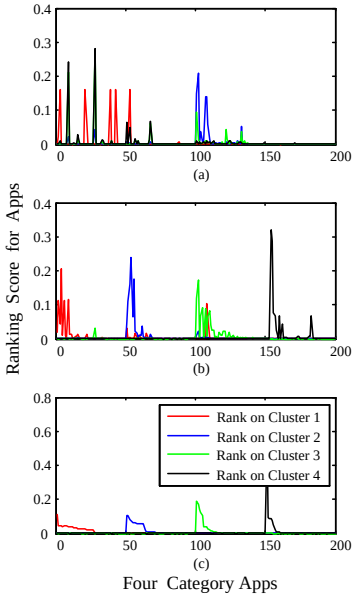

Figure 2: Improvement of Ranking through Iteration

Table 3 shows the top10 companies in the four clusters after iterative computations, from this table, the ranking algorithm has good effect, however this ranking result is different from popularity ranking result. For Tools/Software cluster, a company named 'Baoruan' has the highest value, 'Baidu' is more famous than this company but rank behind this company. After analyzing the results, we find that although 'Baidu' release many popular Apps, only part of these Apps related to other categories. This led to the ranking of 'Baidu' below 'Baoruan' and other 'single-minded' companies.

4.4 Clustering Effectiveness Evaluate

Table 4 reports the Apps clustering results, in which *precision* and *recall* are employed as the metrics. It can be

Table 4: Performance of Apps Clustering

DataSet	Precision%				Recall%			
	G	T/S	V/I	R/L	G	T/S	V/I	R/L
Data1	78.3	70.2	84.4	68.4	75.1	73.2	61.1	80.7
Data2	85.6	77.1	95.3	74.7	81.1	79.3	69.1	88.6
Data3	86.4	82.3	96.1	75.9	83.3	80.3	74.4	89.4

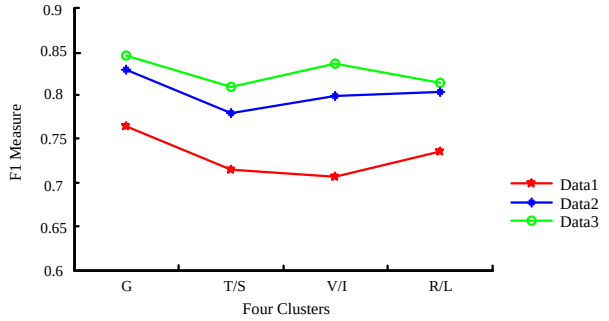

Figure 3: Improvement of Ranking through Iteration

discovered that the clustering performance on Dataset3 is the best. That is to say, the more description information, the better clustering performance. Besides, we notice that the third cluster(V/I) has the lowest recall rate and highest precision rate as well. The forth cluster(R/L) has the highest recall rate and lowest precision rate as well. Figure 3 shows the F_1 values of clustering results on Data1, Data2 and Data3. The results of Game cluster are better than other clusters, even though it's precision or recall is not the highest. This is because the recognition of the Game Apps is more stronger than other Apps, for example, in Table 2 we distribute Apps which are labeled 'Children/Parenting' category to Reading/Learning cluster, however some Apps labeled 'Children/Parenting' can also belong to Tools/Software cluster, but in the process of the assessment, we identify them only belong to Reading/Learning cluster, this is caused by the characteristic of Apps data. We draw two conclusions from the experiment: (1) the wider the difference between the clusters is, the more effective the clustering will be; (2)the purer the cluster is, the more effective the clustering will be.

5. CONCLUSION AND FUTURE WORK

With the development of mobile Internet, data analysis on mobile Apps data attracts more and more attention. Many data mining tasks have been exploited in mobile Apps data, among which clustering and ranking are two fundamental tasks. Most existing algorithms utilize only one or two types of information, while in the paper we propose to cluster and rank mobiles Apps based on a heterogeneous information network, which models networked data as networks including different types of objects and relations. In order to make ranking and clustering mutually enhance each other while analyzing the information network, we introduce a ranking-based clustering algorithm and make it suitable for the mobile Apps scenario. To evaluate the performance, we implement the proposed approach based on a real-world large-scale dataset which contains more than 500,000 Apps. In our future work, we will try to add more Apps related social information into the proposed approach to further improve the clustering & ranking performance.

6. ACKNOWLEDGMENTS

This research was partially supported by the National Technology Support Program under grant of 2011BAH16B04, the National Natural Science Foundation of China under grant of 61173176, Science and Technology Program of Zhejiang Province under grant of 2013C01073, National High- Tech Research and Development Plan of China under Grant No. 2013AA01A604.

7. REFERENCES

[1] A. Banerjee, S. Basu, and S. Merugu. Multi-way clustering on relation graphs. In *SDM*, volume 7, pages 145–156. SIAM, 2007.

[2] A. Clauset, M. E. Newman, and C. Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[3] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[4] B. Long, X. Wu, Z. M. Zhang, and P. S. Yu. Unsupervised learning on k-partite graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 317–326. ACM, 2006.

[5] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[6] C. Shi, R. Wang, Y. Li, P. S. Yu, and B. Wu. Ranking-based clustering on general heterogeneous information networks by network projection. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 699–708, 2014.

[7] Y. Sun, C. C. Aggarwal, and J. Han. Relation strength-aware clustering of heterogeneous information networks with incomplete attributes. *Proceedings of the VLDB Endowment*, 5(5):394–405, 2012.

[8] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu. Rankclus: integrating clustering with ranking for heterogeneous information network analysis. *Proceedings of the 12th International Conference on Extending Database Technology (EDBT)*, pages 565–576, 2009.

[9] Y. Sun, B. Norick, J. Han, X. Yan, P. S. Yu, and X. Yu. Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1348–1356. ACM, 2012.

[10] Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 797–806, 2009.

[11] R. Wang, C. Shi, S. Y. Philip, and B. Wu. Integrating clustering and ranking on hybrid heterogeneous information network. In *Advances in Knowledge Discovery and Data Mining*, pages 583–594. Springer, 2013.