Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 601528, 11 pages
http://dx.doi.org/10.1155/2013/601528

Research Article

KaM_CRK: Clustering and Ranking Knowledge for Reasonable Results Based on Behaviors and Contexts

Changhong Hu,[1,2] Shufen Liu,[1] Ramana Reddy,[2] Sumitra Reddy,[2] and Mingyang Liu[1,2]

[1] *College of Computer Science and Technology, Jilin University, Changchun, Jilin 130012, China*
[2] *KaMLab, Lane Department of Computer Science & Electrical Engineering West Virginia University, Morgantown, WV 26506, USA*

Correspondence should be addressed to Changhong Hu; hu940mr@gmail.com

Received 20 January 2013; Accepted 5 May 2013

Academic Editor: Jun Zhao

A model named KaM_CRK is proposed, which can supply the clustered and ranked knowledge to the users on different contexts. By comparing the attributes of contexts and JANs, our findings indicate that our model can accumulate the JANs, whose attributes are similar with the user's contexts, together. By applying the KaM_CLU algorithm and Centre rank strategy into the KaM_CRK model, the model boosts a significant promotion on the accuracy of provision of user's knowledge. By analyzing the users' behaviors, the dynamic coefficient Behavior$F$ is first presented in KaM_CLU. Compared to traditional approaches of K_means and DBSCAN, the KaM_CLU algorithm does not need to initialize the number of clusters. Additionally, its synthetic results are more accurate, reasonable, and fit than other approaches for users. It is known from our evaluation through real data that our strategy performs better on time efficiency and user's satisfaction, which will save by 30% and promote by 5%, respectively.

1. Introduction

With the swift development of computer technology, especially the development of computer networking, the internet has indeed changed the habit of people searching for knowledge. People commonly use Google and Facebook as their search engines to find out the knowledge they need. The search engines often supply the service or knowledge by users' searching keywords. As a matter of fact, people more wish to know more related information about the keywords based on users' context. So we built a researching knowledge system whose name is Knowledge advantage Machine (KaM). KaM focuses on binding ontology pattern from user activities and discovering knowledge based on users' context. Since we all know that there are tons of algorithms and resolutions on discovering knowledge in our modern world, so in this system the key and most crucial problem is how to present the proper clustered and ranked knowledge to the users based on users' contexts that can save users' searching time and make the knowledge acquisition easier.

Predecessors in this research area have already constructed some models, algorithms, and methods, which have been implemented in high-speed internet, knowledge research, and information searching engines. While among these algorithms and models, the most well-known and classic ones are PageRank, HITS, K_means, and DBSCAN. PageRank is a link analysis algorithm that is named after Larry Page [1] and used by the Google search engine. The search engine's main principle locates and assigns a numerical weighting to each element of a hyperlinked set of documents, such as the WorldWideWeb, with the purpose of "measuring" its relative importance within the set (Figure 1).

Hyperlink-Induced Topic Search (HITS) also known as hubs and authorities developed by Jon Kleinberg is a link analysis algorithm that rates web pages [2]. It was a precursor to PageRank. The idea behind hubs and authorities stemmed from a particular insight into the creation of web pages when the Internet was originally forming; that is, certain web pages, known as hubs, served as large directories that were not actually authoritative in the information that it held,

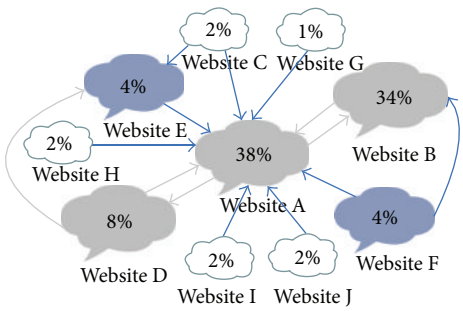

Figure 1: PageRank.

but were used as, compilations of a broad catalog of information that led users directly to other authoritative pages. In other words, an efficient hub represented a page that pointed to many other pages, and an efficient authority represented a page that was linked by many different hubs.

K_means algorithm needs to input quantity of clusters k. Then n data_objects are divided into k clusters. The similarities of objects are bigger if they were in the same cluster, otherwise, that would be smaller. The more different the clusters were, the less similar they would be. Cluster similarity is related to the mean of clusters gaining the center of gravity to carry out the calculation [3–5].

Density Based Spatial Clustering of Applications with Noise (DBSCAN) is an algorithm based on the density of clustering. It is different from the division and hierarchical clustering method. It will define the cluster as the point of the biggest density connected set. Clusters are divided into areas according to the threshold of their density. In the spatial data, where clustering the noise arbitrary shape can be found [3–11].

By using the number of web page links, those former algorithms have already resolved the ranking and clustering problems, after which the results will return back by search engine mechanically, that certainly ignores our human beings' real needs at some point. In other words, what the knowledge really focuses on is users, and users are host of the knowledge, who will be the terminal nodes of knowledge usage. All the knowledge and information would be meaningless if there were not users involved. Without users' behaviors and contexts we consider that the results after ranking and clustering cannot fulfill the users' requirements.

So we put forward a framework called KaM_CRK shortly, which concentrates on users' behaviors and contexts. In KaM we propose a definition of "JAN," which is an abstract object for all the general knowledge resources. It provides an abstract definition about the knowledge to achieve the uniqueness for different users using the same resources. So we formed a second definition which describes the attributes of JANs.

By comparing the users' contexts and JANs attributes, we can accumulate JANs which have similar attributes as the user's context attributes. The definition of "JAN" is given in Section 2. We also introduced the users' behaviors coefficient Behavior$F$ into KaM_CRK, by adopting which the JANs are ranked and clustered following users' behaviors, JANs attitudes, and context attitudes. The following is the main contributions of our work.

(a) We build concepts of context attributes and JANs attributes, whose center ideas are that different contexts come up with different attributes and one JAN also has multiple attributes.

(b) We present an algorithm named KAM_CLU which is the root from universal gravitation. We assign the different definitions of the coefficients in KAM_CLU. The coefficients will be calculated by the users' behaviors and context.

(c) We formally bring up a Centre_rank strategy which is for ranking clusters reasonably. This strategy mixes the users' behaviors and JANs_Distance method. It leads the satisfactions of users on JANs' ranking with an increase of 5%.

(d) At last we also perform an experiment using synthetic data and real data from Gavin and Nad. The results demonstrate that KAM_F and Centre_rank can cluster and rank the knowledge dynamically and accurately. The users' satisfactions are better than the traditional methods such as K_means and DBSCAN.

The rest of paper is organized as follows. Section 2 is on related work and the former research of our team. In Section 3, we prepare a set for Section 4 and next sections. We introduce a cluster method named KaM_CLU and how to get the coefficients of KAM_F in Section 4. In Section 5 a Centre_rank strategy is presented. We make an analysis on our experimental results in the last two sections.

2. Related Work

Since our team does research on network data mining, we have accumulated some knowledge about the related fields such as the most famous algorithms of PageRank [1] and HITS [2]. Besides, we discovered some improved algorithms which were RankCompete [12] and RankClus [13]. In the following, not only the differences between KAM_CRK and former algorithms are discussed, but also those and models of our previous research are presented.

As that is known, PageRank is used on Google which does ranking and clustering through Hyperlinks in web page. The scores of web page on the network are calculated by importance of Hyperlink on other websites. If the web page has a Hyperlink from another web page, it will get one score. If there is no Hyperlink on this web page, the score of this web page is empty. The way to score one web page on the internet relies on the importance of in and out Hyperlinks within it. The rule is that only the web page will get score when there is one Hyperlink connected to it. Analogously, the more Hyperlinks do the web pages have, the more scores it will get. Instead, the web page will be scored zero if there is no Hyperlinks related to itself. To cluster and rank the knowledge are based on the Hyperlinks from other knowledge than itself. And the HITS is another improved algorithm of PageRank which uses Hub Scores

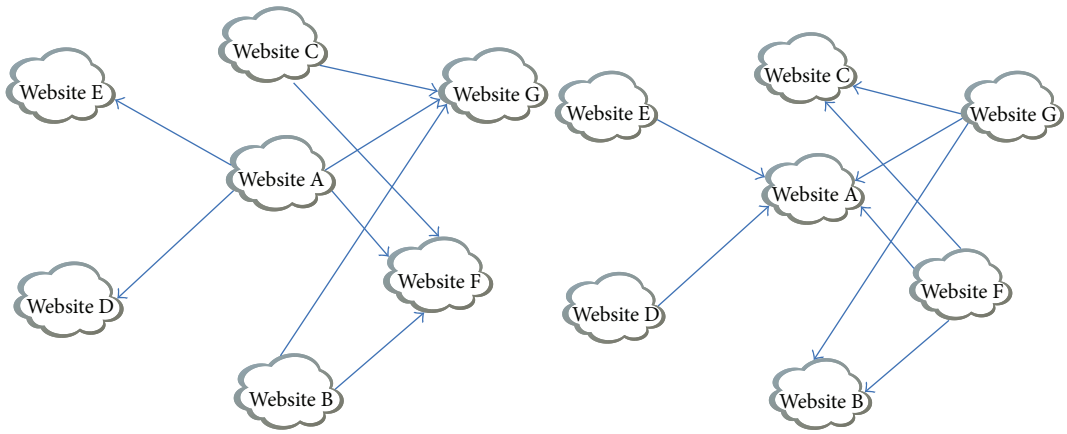

Figure 2: HITS.

and Authority Scores to rank and cluster knowledge (Figure 2). Hub Scores of knowledge are the total Authority Scores which link the same knowledge. Authority Scores of knowledge are total Hub Scores which link in the same knowledge. RankClus is an algorithm which is proposed by Suny et al. They use a mixture model to decompose each knowledge 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. Although RankClus has resolved the problems of ranking and clustering in the meantime, with no regard to the relationship between the knowledge and users' context, which is useless and meaningless; Jeh and Widom proposed a RankCompete algorithm at the University of Illinois [14]. The unique advantage of this algorithm allows multiple random walkers one time in the same network existing work mostly focus on random walks of a single category. By introducing the concept "competition" into the random walk framework, their method can fulfill both clustering and ranking knowledge simultaneously in order further to provide an effective analysis tool for networks. It cannot be denied that RankCompete is a very good approach. But it neglects the users' experiences. Every person has his own attitudes of behaviors; we should offer the proper knowledge based on our own behaviors not just using the knowledge's competition to make a users' decision.

In 2011, our team published 2 papers which are indexed by SCI about clustering, ranking, and grouping whose titles are "*A universal gravitation based clustering algorithm for distributed file system*" [15] and "*group competitive model of optimal node selection based on service evaluation*" [16]. The first paper describes a universal gravitation based clustering algorithm. This algorithm adopts the law of universal gravitation, which gives strategies for node movement. Meanwhile, to overcome premature or local-best solution, the theory of overcoming prematurity is referred, and then node can depart for a more suitable cluster. Theoretical proof shows the algorithm is of convergence and has the top limit in the time complexity. The second paper presents a grouping model in which there is a ranking algorithm. This paper's main idea is that from the evaluations of all the nodes we can find the best node for the user.

The above papers can resolve the related problems of ranking and clustering. However the concentrations of the algorithms depend all on the evaluations of knowledge themselves instead of taking users into account. Users will use the ranked knowledge in the end, so in this paper we present an idea whose central method is all around users' behaviors and contexts.

3. Problem Formulation

In the real word, users have different needs for different contexts: for example, in a (computer themed) conference where an author needs the computer science knowledge such as computer papers or books to present and communicate with other scholars. So under such a context this author has a computer attribute. In this paper we compare different attributes of different contexts with the attributes of knowledge about different users to cluster and rank knowledge.

Definition 1 (attributes on different contexts). Given a type of object set X, where $X = \{x_1, x_2, \ldots x_m\}$, set X is called an attribute set, if a user is in a context, who must have an attribute in set X. $x_1, x_2, \ldots x_m$ are the attributes of a user in different contexts.

Figure 3 is an example of attributes, in which there are three attributes which are meeting attribute, work attribute, and shopping attribute on three contexts. So we can simply conclude that three attributes exist in this user's attribute set.

Definition 2 (attributes of different JANs). Given a type of JANY set Y, where $Y = \{y_1, y_2, \ldots y_n\}$, set Y is called a JAN's attribute set, if a JAN has different attributes, which must be in set Y.

Figure 4 is an example of JAN attributes. It shows that a paper has 4 different attributes which are computer, math, medical science, and bioinformatics.

(1) Input: user's context attribute $A \in$ *user's* context attribute set XA and JANs' attribute sets $Y_1, Y_2, \ldots, Y_k$ and $y_i \in Y_1 \cup Y_2 \cup \cdots Y_k$. The number of elements in $Y_1 \cup Y_2 \cup \cdots Y_k$ is m.
(2) Initialize: Select attribute A to compare with elements of $Y_1, Y_2, \ldots, Y_k$ sets.
(3) If $A \sim y_i$, put $y_i \leftrightarrow$ JAN in Prepare Set A and $i++$.
(4) If $i \neq m$ continue and Update Set A.
(5) return (3)
(6) if $i = m$
(7) end if.
(8) Output: Prepare Set A.

Algorithm 1: Prepare Set.

(1) Input: Prepare Set A, JANx_{weight}, JANy_{weight}, JANs$_{\text{Distance}}$.
(2) Initialize: Select elements from Set A to compare with each other.
(3) KAM_F = Behavior$F$ $\left(\text{JAN}x_{\text{weight}} \times \text{JAN}y_{\text{weight}} / \text{JANs}^2_{\text{Distance}}\right)$ find the gravitational JANs and cluster together $\{\text{Cluster}_1, \text{Cluster}_2, \ldots, \text{Cluster}_k\}$.
(4) Output: $\{\text{Cluster}_1, \text{Cluster}_2, \ldots, \text{Cluster}_k\}$.

Algorithm 2: Cluster.

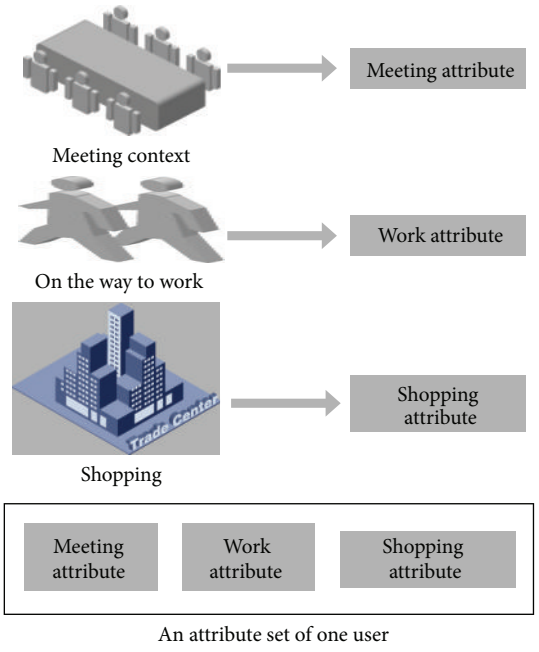

Figure 3: Attribute set on contexts.

KaM_CRK is an idea about context and users. In our approach we build relationships between them which are described by an algorithm of "Prepare Set." It compares attributes of user's context with elements of JANs' attribute sets. If we find one element of JANs' is similar to the user's context attribute, we will pick up the corresponding JAN and put it into the Prepare Set. The Prepare Sets are treated as the nodes of clustering in Algorithm 1.

In the next words we will use an example to explain the Prepare Set such as Figure 5. UserA has three contexts which are context1, context2, and context3, respectively. JANx, JANy, and JANz separately have attribute1, attribute2, attribute3, attribute4, attribute5, attribute6, attribute7, attribute8, and attribute9. Meanwhile context1, context2, and context3 have the similar attributes with attribute1 to attribute9, respectively. So we put the corresponding JANs together as new sets which are called Prepare Sets. Every Prepare Set is deemed to a new JAN or a new node. This algorithm is a preparation for the next stage of clustering.

4. Clustering

In this section, we introduce a cluster method KAM_CLU to cluster the Prepare Set in Section 3 based on behaviors and contexts. In context human beings often use JANs whose attributes are very similar to the context attribute. In the perspective of some users' behaviors sometimes they need not only the related JANs but also the irrelevant JANs. Let us take this, for instance, that a Ph.D. candidate named David whose major is computer. When it comes to preparing a presentation, we would normally consider that it is related to his major that computer science. But in fact, we should take the consideration of his behavior is that he may require the domain of physics knowledge more often. So in this algorithm we should both consider the similarity of attitude and the users' behaviors. And in KAM_CLU we introduce the factor Behavior$F$ into this algorithm.

Algorithm 2 finds the JANs with similar attributes will be clustered together. The following is the equation of KAM_CLU:

$$\text{KAM}\_F = \text{Behavior}F \frac{\text{JAN}x_{\text{weight}} \times \text{JAN}y_{\text{weight}}}{\text{JANs}^2_{\text{Distance}\,xy}}. \quad (1)$$

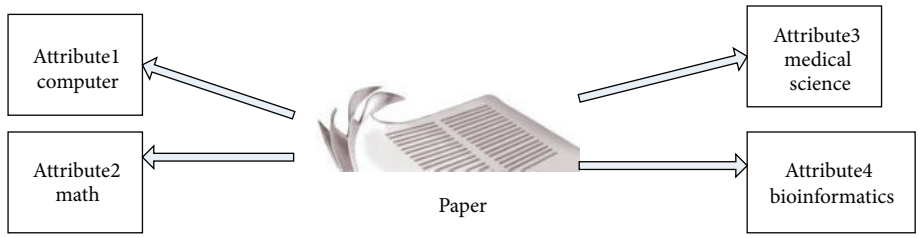

Figure 4: Attributes of JAN.

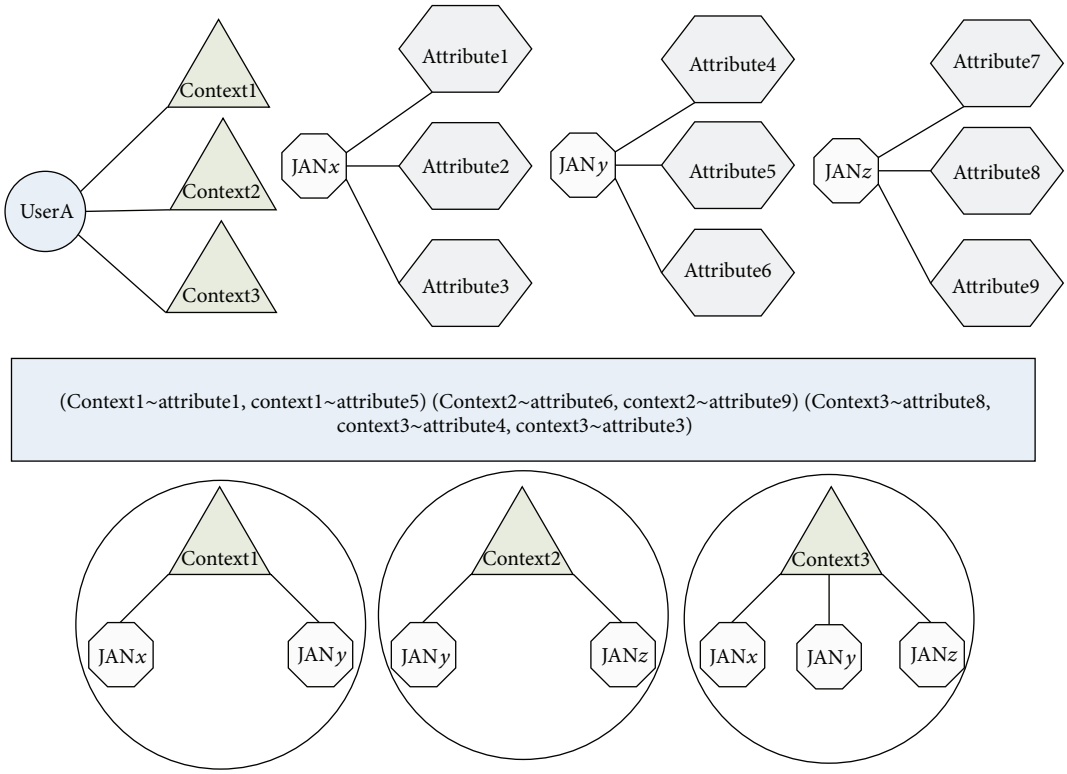

Figure 5: Prepare Set.

It can be easily seen that the formulation in KAM_CLU is very similar to the Universal Gravitation Equation, except for that which have the different the coefficients and definitions.

4.1. KAM_CLU Strategy. In KAM_CLU there are two strategies: the gravitational strategy and the mutually exclusive strategy.

Figure 6 is an example of the mutually exclusive strategy. When the destination node "Medical Paper" has not the similar attribute node of the source node, the mutually exclusive strategy is triggered. The internal force and the external force are in opposite directions. The "Medical Paper" is far from the computer paper cluster.

Figure 7 is an example of gravitational strategy. When the destination node computer paper 6 is the closed node of the source node computer paper, the gravitational strategy is triggered. The computer paper 6 becomes a node of computer paper cluster.

In our former work [15] we have proved the strategy convergence. So in this paper we will not make an adequate account.

4.2. BehaviorF in KAM_CLU. As it is known, a kind of universal gravitation is a law which is the existing objective with the character of moving and material. Based on this point, the gravity coefficient will be changeable as the passage of time and space transformation. Normally, when it comes to the calculation of the coefficient, the universal gravitation G can be concerned as a constant for further calculating.

Specifically when it comes to KAM_CLU, we consider the coefficient Behavior$F$ to be dynamic instead of being static, which will be acquired by calculating users' behaviors. We are also inspired by Latent Semantic Analysis (LSA) [3, 4], which analyzes relationships between a set of documents and the terms. But driven by that, we build the relationships of JANs from our analysis of the users' actions.

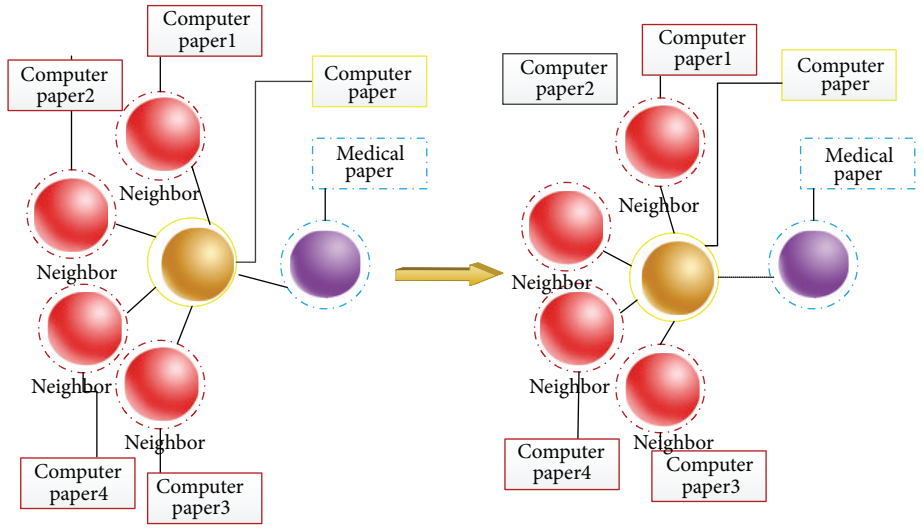

Figure 6: Mutually exclusive strategy.

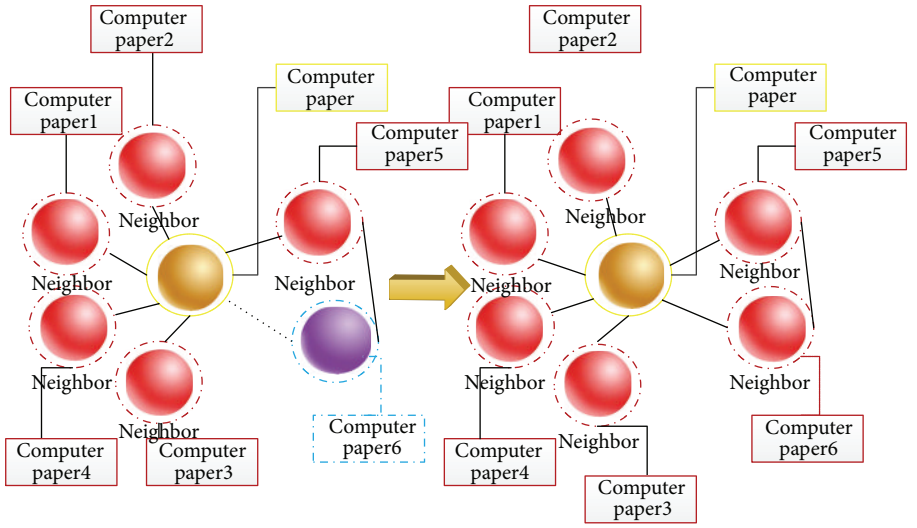

Figure 7: Gravitational strategy.

At the beginning, we build a training machine and divide the training time into pieces. Every piece is called Time Range (TR).

Definition 3 (TR). Given a set of $\mathrm{TR}_{\mathrm{SET}} = \{\mathrm{TR}_1, \mathrm{TR}_2 \ldots \mathrm{TR}_n\}$ which is a piece time set, $\mathrm{TR}_1 = \mathrm{TR}_2 = \cdots = \mathrm{TR}_n$, and $\mathrm{ALL}_{\mathrm{TIME}} = \sum_{i=1}^{n} \mathrm{TR}_i$. From recording the users' selections of any two JANs appearing together in TR_i that will be put times of selections in a Times_together set. We also insert the selections' frequencies in $\mathrm{TS}_{\mathrm{matrix}}$.

Definition 4 (Times_together set). Hypothetically, the $\mathrm{TST}_{\mathrm{SET}}$ number of JANs is N_{JANs}, and given a set in which there are the times of any two JANs selection together,

$$\mathrm{TST}_{\mathrm{SET}} = \left\{ \mathrm{JAN}_{x-y}\mathrm{TR}_1, \ldots, \mathrm{JAN}_{x-y}\mathrm{TR}_i \mid 1 < i < C^2_{N_{\mathrm{JANs}}}, \right.$$
$$\left. x \neq y, 1 < x < N_{\mathrm{JANs}}, 1 < y < N_{\mathrm{JANs}} \right\}. \quad (2)$$

Definition 5 (times matrix). Given a $i \times a$ matrix $\mathrm{TS}_{\mathrm{matrix}}$ in which there are the times about separate selections of JANa, $1 < a < N_{\mathrm{JANs}}$. Consider

$$\mathrm{TS}_{\mathrm{matrix}} = \begin{bmatrix} \mathrm{JAN}1_{\mathrm{TR}_1} & \cdots & \mathrm{JAN}a_{\mathrm{TR}_1} \\ \mathrm{JAN}1_{\mathrm{TR}_2} & \cdots & \mathrm{JAN}a_{\mathrm{TR}_2} \\ \vdots & & \vdots \\ \mathrm{JAN}1_{\mathrm{TR}_i} & \cdots & \mathrm{JAN}a_{\mathrm{TR}_i} \end{bmatrix}, \quad (1 < a < N_{\mathrm{JANs}}). \quad (3)$$

We use the least square method to curve fitly the set $\mathrm{TST}_{\mathrm{set}}$ and get an equation $P_l(x) = \sum_{k=0}^{l} a_k x^k$. Then we can calculate the entropy of $p_l(x)$:

$$H\left(P_l(x)\right) = E\left(P_l(x)\right),$$
$$H\left(P_l(x)\right) = \sum_{I=1}^{k} p\left(x_I\right) I_l\left(x_I\right),$$

$$H\left(P_l\left(x\right)\right) = -\sum_{I=1}^{k} p\left(x_I\right)\log\left(p\left(x_I\right)\right),$$

$$\text{Behavior}F = \frac{-\sum_{I=1}^{k} p\left(x_I\right)\log p\left(x_I\right)}{\left(\sum_{m=1}^{i}\left(\left(\text{JAN}x_{\text{TR}_m}\right) + \text{JAN}y_{\text{TR}_m}\right)\right) \div i}. \tag{4}$$

In this paper Behavior$F$ is not a static number. It is fluctuating with TST_{SET}. With the number of elements in TST_{SET} increasing, the samples of training data are more sufficient. Then the coefficient Behavior$F$ is more accurate.

4.3. JAN_{weight} Based on Feedback. In this section our approach is that by analyzing the users' feedback we will evaluate the scores based on different JANs attributes. We treat the scores as the $\text{JAN}_{\text{weight}}$. In Definition 2 we have defined attributes of different JANs. $\text{JAN}_{bs} = \text{JAN}_{\text{weight}}$ is the JANs' attributes basic score. So in this section the number of subsets in JAN_{bs} is the same as the number of JANs' attributes.

Definition 6 (JANs_score). Given a set named JAN_{bs} whose subsets' number is the same as the number of JANs (N_{JANs}), the number of elements in subsets is determined by number of attributes of JANs:

$$\text{JAN}_{bs} = \left\{\left\{y[k,1]_{\text{score}}, y[k,2]_{\text{score}}, \ldots, y[k,n]_{\text{score}}\right\} \mid \left(1 < K < N_{\text{JANs}}\right)\right\}. \tag{5}$$

n is determined by the attributes of different JANs. In this paper we use the short-term learning of the relevance feedback algorithms to analyze and record the scores in JAN_{bs}. The details of this algorithm are in [17].

4.4. JANs_Distance Constrained by Attributes Transferred. In this section we introduce an idea of JANs_Distance. JANs_Distance is a logical distance, a semantic distance, and a context distance rather than physical distance. For example, a JAN of computer paper has different attributes. When a user is in different contexts, he will use the different attributes of JANs. Figure 4 is an example about attributes of a JAN. But these JAN's attributes are all different among each other. So we should assign the different coordinates to different attributes such as Figure 8. Math has the closer relationship with computer. So they are very close to each other. If the two attributes had no closer relationship, the distance would be bigger.

If the user's major is about computers, he will use the JAN's attribute about computers. We supply the user this JAN's computer attribute coordinate.

In this paper the semantic or JANs' attributes coordinates assignment is not the main discussion. In today's academic circles there are still some mature algorithms which can resolve this problem such as Edge Counting Measures, Information Content Measures, Feature-Based Measures, and Hybrid Measures. So, we do not need to exemplify the coordinates of semantic.

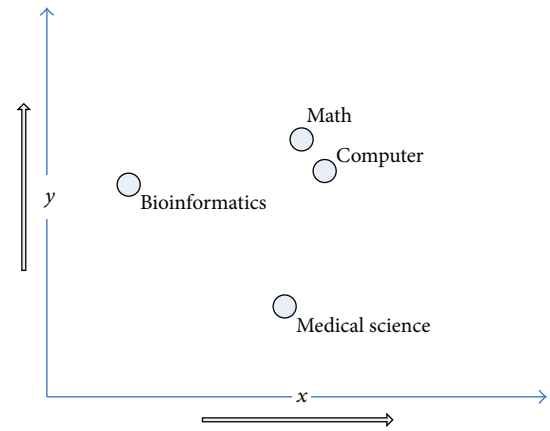

Figure 8: One JAN's attributes coordinates.

Definition 7 (JANs' attributes coordinates). Given a JANs' attributes coordinates set which is named $\text{JAN}_{\text{coordinate}}$, the number of subsets is the same as the set JAN_{bs}:

$$\text{JAN}_{\text{coordinate}} = \left\{\left\{J_c\left[j,1\right], J_c\left[j,2\right], \ldots, J_c\left[j,n\right]\right\} \mid \left(1 < j < N_{\text{JANs}}\right)\right\},$$

$$\text{JANs}_{\text{Distance}} = \left|J_c\left[W_{x1}, W_{y1}\right], Jc\left[W_{x2}, W_{y2}\right]\right| \times \left(1 < W_{x1}, W_{y1} < n, 1 < W_{x2}, W_{y2} < N_{\text{JANs}}\right). \tag{6}$$

5. Centre_Rank Strategy

As we all know that the most widely used ranking methods in modern academic community are all based on authority Ranking. But in this section we introduce a different ranking strategy. This strategy is based on users' behavior. First we use the training machine to record the behaviors of users' selecting JANs'. Then we use the law of entropy to calculate the users' behavior JAN's coordinate. Thus we can get a coordinate of the user's most favorite attribute. We will treat the JAN_{bs} as the origin of a circle. And the distance between coordinate of the user's most favorite attribute and the JANs_score is the radius. These results leave us with a circle in which there are some JANs' attributes. We should find which cluster's elements are the most and treat this cluster as the 1st. And we increase the radius return to the above procedures and find the second. Iteratively we do this loop until we scan all the clusters.

Figure 9 is an example of Centre_rank. In this picture circle1 is a circle whose radius is based on the user's behaviors. We can see that cluster1, cluster2, and cluster3 are all in circle1. Although cluster1 and cluster2 are closer to the original JAN_{bs}, the JANs of cluster 3 in this circle are more than other clusters. So the sequence is cluster3, cluster2, and cluster1. And then we increase the radius and get the circle2. From the strategy we can get the rank is cluster4, cluster5.

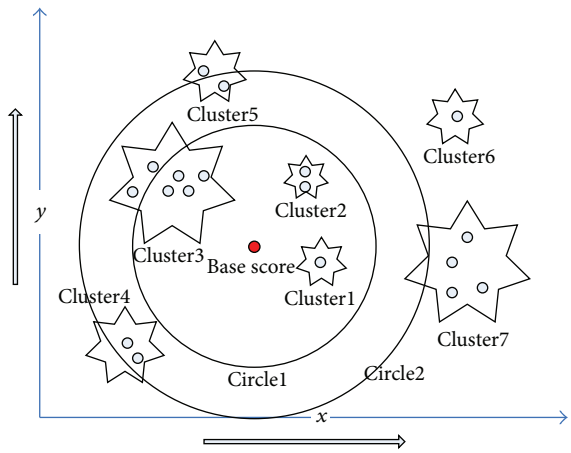

Figure 9: Centre_rank.

The total rank is cluster3, cluster2, cluster1, cluster4, cluster5, cluster7, and cluster6.

6. Experiments

In this section, we will show the effectiveness and the satisfaction of users. The effectiveness of synthetic results will be discussed in Sections 6.1 and 6.2. The satisfaction of users will be discussed in Section 6.3 which will use the real data by volunteers.

6.1. Synthetic Results. In this paper we will pick up two traditional algorithms for comparing accuracy with KaM_CRK. The two algorithms' names are K_means and DBSCAN. In KaM_CRK we do not need the parameters to configure the algorithm. But in the other two traditional algorithms we need to define the parameters.

In this experiment we first build 3 groups data.

Group 1:

Number of samples = [20, 20, 20].

Number of K_means clusters = 7.

Scan radius of DBSCAN = 0.3.

Minimum contains points of DBSCAN = 2.

Group 2:

Number of samples = [50, 50, 50].

Number of K_means clusters = 14.

Scan radius of DBSCAN = 0.3.

Minimum contains points of DBSCAN = 2.

Group3:

Number of samples = [150, 150, 150].

Number of K_means clusters = 40.

Scan radius of DBSCAN = 0.3.

Minimum contains points of DBSCAN = 2.

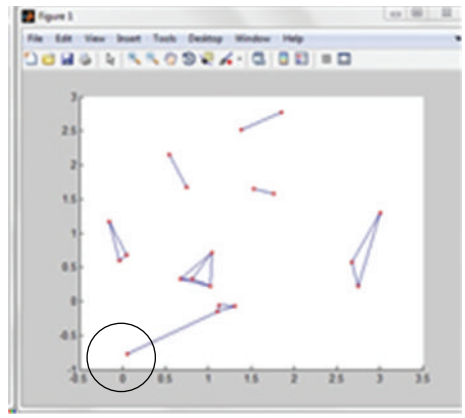

Figure 10: KaM_CRK Group1.

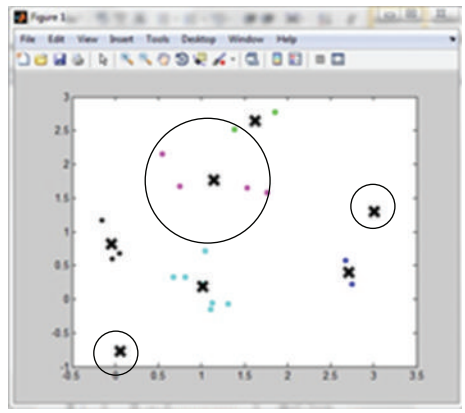

Figure 11: K_means Group1.

Experiments' Environment. The algorithm is compiled using MATLAB 7.9.0. Runtime environment is Windows 7 SP1 64 bit, Intel(R) i5 CPU, 8.00 GB RAM.

Figure 10 is the result of KaM_CRK. It can easily be found that in this picture there is no one-node cluster. Every cluster at least has two elements. The circle marked node is very far from the cluster, but it is still concluded in the cluster. The reason is the node is controlled by the user's behaviors. The coefficient Behavior$F$ plays a leading role.

Figure 11 is the result of K_means which uses Group1 samples. We can find that the distributions of clusters are different from Figure 10, in which has two one-node clusters, which are marked by small circles. There still are four disperse nodes which are put in one cluster and marked with big circle.

Figure 12 shows a result of KaM_CRK using Group2 data. There is also no one-node cluster. The number of clusters is thirteen.

We can find that there are still some one-node clusters in Figure 13. With the number of samples increasing, the number of one-node clusters is increasing.

Figure 14 presents that there is only one one-node cluster. And the number of clusters is forty.

In Figure 15 we can find out seven one-node clusters.

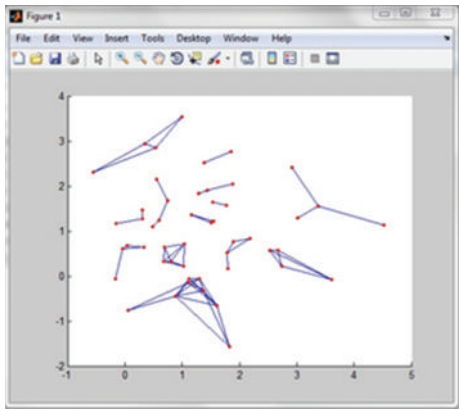

Figure 12: KaM_CRK Group2.

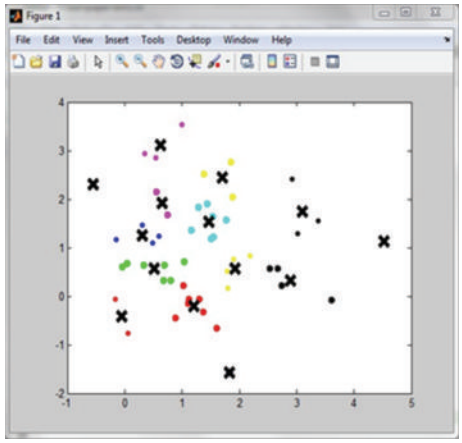

Figure 13: K_means Group2.

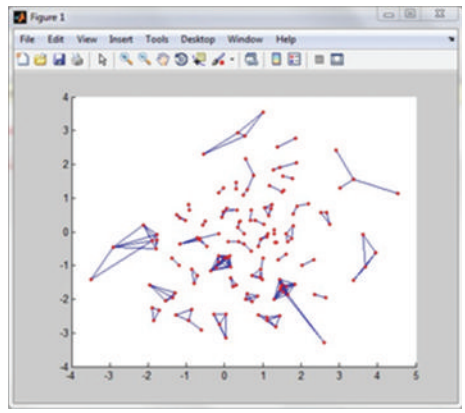

Figure 14: KaM_CRK Group3.

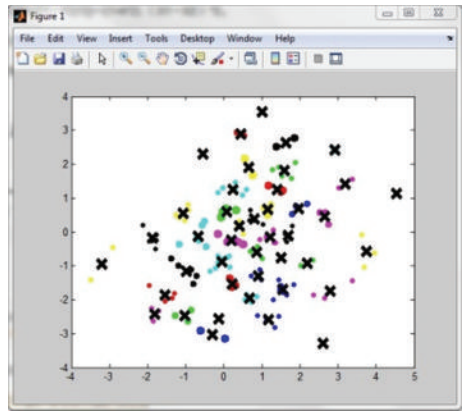

Figure 15: K_means Group3.

6.2. Synthetic Results Discussion. From the last section we can conclude that KaM_CRK can cluster the JANs very well. In this section we will use the KaM_CRK results to compare with the other traditional algorithms.

In Figure 16 we can see that the numbers of clusters about KaM_CRK and K_means are the same, which is bigger than DBSCAN. From our analysis we consider that KaM_CRK needs not to configure the parameters of number of clusters. The number of clusters is built by synthetic data. But with K_means we must take the number of clusters to the algorithm. So from our view the KaM_CRK is more reasonable than others such as K_means, DBSCAN and so on. And we also suggest that DBSAN does not fit the accurate cluster. It can be used for the density clusters. KaM_CRK considers the users' contexts and behaviors in the algorithm. We consider that KaM_CRK's results are accurate.

From Figure 17 we can get that using our approach the number of one-node clusters is less than others such as K_means, DBSCAN and so on. When the number of JANs is 20, the number of one-node clusters is 0. But the number of one-node clusters using K_means and DBSCAN is separately 3 and 10. The number of one-node clusters is still 0, when the number of JANs is increasing to 50 in KaM_CRK. Even if the number of JANs is 150, the number of one-node clusters is only 1. But after K_means cluster 50 JANs, there are still 3 one-node clusters existing, and DBSCAN's one-node cluster is rising to 24. With the number of JANs increasing to 150, K_means' number of one-node clusters is a small increase which is 7. But the number of JANs is still 150, and the number of one-node clusters is dropping to 3 using DBSCAN.

From the above results and our analysis, the effectiveness of clustering using KaM_CRK is better than others such as K_means, DBSCAN, etc. Few one-node clusters mean that the users can save time to collect proper JANs from all the knowledge. Maybe some scholars consider that when the number of clusters is bigger, the effectiveness is worse. But we regard that our approach is based on users' behaviors and contexts; though the number of cluster is a little bigger, the clustering of our method is more accurate. The accurate clusters can save the users' time. For a large number JANs, using the DBSCAN algorithm is also a good idea, because with the JANs size increasing, the number of one-node clusters has an apparent drop. The reason is DBSCAN treats the one-node clusters as noise. So when the nodes are intensive, the number of one-node clusters decreased. But DBSCAN and K_means have no relationship about users' behaviors and contexts.

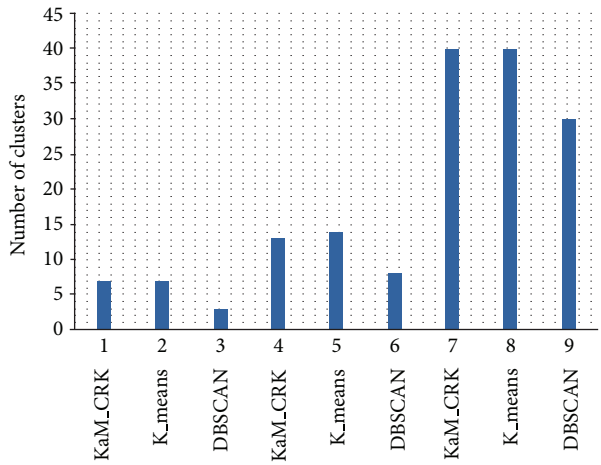

Figure 16: Number of clusters.

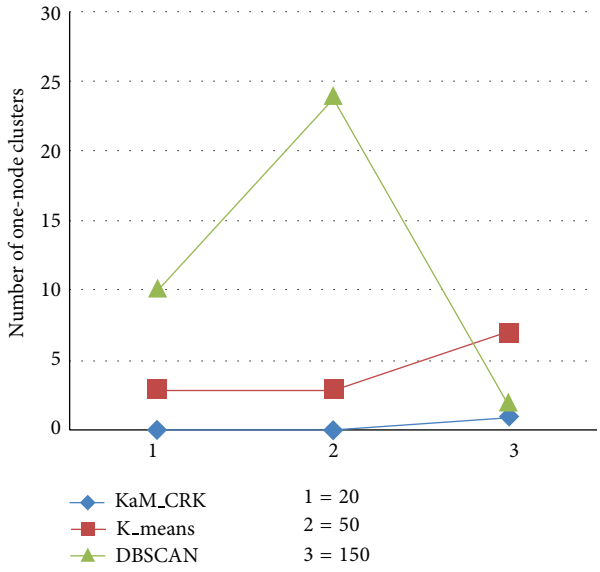

Figure 17: Number of one-node clusters.

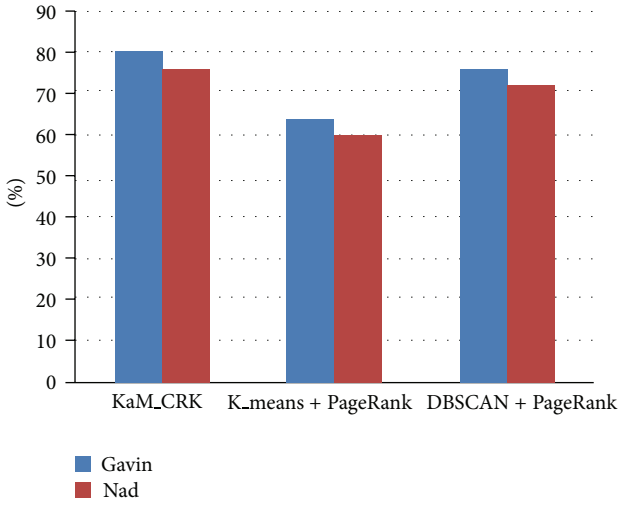

Figure 18: Gavin's and Nad's satisfaction.

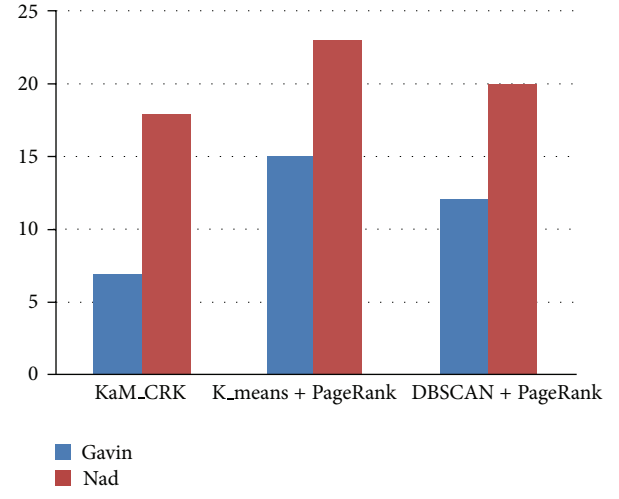

Figure 19: Gavin's and Nad's cost time.

6.3. Real Results and Discussion. Two volunteers from West Virginia University whose names are Gavin and Nad and whose majors are computer and sport, separately cooperate with our experiment. Gavin supplied us with 200 JANs such as super links, papers, PPTs, and photos, which are all about computers. Nad supplied us with 170 JANs, which are about sport. We use these real data on KaM. We supply the real clustered and ranked results to Gavin and Nad. They gave the evaluations back to us.

Figure 18 is the satisfaction of Gavin and Nad. We can easily find that KaM_CRK's result is better than K_means + PageRank and DBSCAN + pageRank. The satisfaction has a 5% increase. From Figure 19 we can see that the cost time of KaM_CRK is still less than other algorithms about 30% decrease. But we discover that Nad's cost time is apparently more than Nad's. After our analysis, the reason is about their majors. Gavin's major is computer which we are familiar with. In the assignment of context attributes and the JANs attributes we can make them accurately. But we are not familiar with Nad's major (sport), so some problems happen on assignment of attributes about JANs and contexts. In the postresearch we will pay more attention on this aspect.

7. Summary

In this paper we introduce users' behaviors and contexts into the clustering and ranking. A KaM_CRK model is built in which there are one algorithm and a strategy. The algorithm's name is KaM_CLU which can cluster the JANs accurately based on JANs attributes and contexts attributes. The strategy's name is Centre_rank which can rank the clusters reasonably. We validate our approach on computer JANs and sport JANs. The effect is better than traditional methods. Our future work is building more contexts attributes and JANs attributes for KaM.

References

[1] "Google Press Center: Fun Facts," http://www.google.com/.

[2] *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[3] T. K. Landauer, P. W. Foltz, and D. Laham, "Introduction to latent semantic analysis," *Discourse Processes*, vol. 25, pp. 259–284, 1998.

[4] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, pp. 1150–1157, September 1999.

[5] Q. Mei, D. Cai, D. Zhang, and C. X. Zhai, "Topic modeling with network regularization," in *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*, pp. 101–110, April 2008.

[6] L. N. Gitlin, W. W. Hauck, L. Winter, M. P. Dennis, and R. Schulz, "Effect of an in-home occupational and physical therapy intervention on reducing mortality in functionally vulnerable older people: preliminary findings," *Journal of the American Geriatrics Society*, vol. 54, no. 6, pp. 950–955, 2006.

[7] L. N. Gitlin, L. Winter, M. P. Dennis, M. Corcoran, S. Schinfeld, and W. W. Hauck, "A randomized trial of a multicomponent home intervention to reduce functional difficulties in older adults," *Journal of the American Geriatrics Society*, vol. 54, no. 5, pp. 809–816, 2006.

[8] A. Giusti, A. Barone, M. Oliveri et al., "An analysis of the feasibility of home rehabilitation among elderly people with proximal femoral fractures," *Archives of Physical Medicine and Rehabilitation*, vol. 87, no. 6, pp. 826–831, 2006.

[9] M. Crotty, C. Whitehead, M. Miller, and S. Gray, "Patient and caregiver outcomes 12 months after home-based therapy for hip fracture: a randomized controlled trial," *Archives of Physical Medicine and Rehabilitation*, vol. 84, no. 8, pp. 1237–1239, 2003.

[10] R. Kuisma, "A randomized, controlled comparison of home versus institutional rehabilitation of patients with hip fracture," *Clinical Rehabilitation*, vol. 16, no. 5, pp. 553–561, 2002.

[11] M. D. Landry, S. Jaglal, W. P. Wodchis, J. Raman, and C. A. Cott, "Analysis of factors affecting demand for rehabilitation services in Ontario, Canada: a health-policy perspective," *Disability and Rehabilitation*, vol. 30, no. 24, pp. 1837–1847, 2008.

[12] L. Cao, X. Jin, Z. Yin et al., "RankCompete: simultaneous ranking and clustering of information networks," *Neurocomputing*, vol. 95, pp. 98–104, 2012.

[13] Y. Suny, J. Hany, P. Zhaoy, Z. Yiny, H. Chengz, and T. Wuy, "RankClus: Integrating Clustering with Ranking for Heterogeneous Information Network Analysis," ACM. EDBT, Saint Petersburg, Russia, 2009.

[14] G. Jeh and J. Widom, "SimRank: a measure of structural-context similarity," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*, pp. 538–543, ACM, July 2002.

[15] S. Liu, H. Leng, and C. Hu, "A universal gravitation based clustering algorithm for distributed file system," *Chinese Journal of Electronics*, vol. 21, no. 1, pp. 28–32, 2012.

[16] S. Liu and C. Hu, "Group competitive model of optimal node selection based on service evaluation," *Chinese Journal of Electronics*, vol. 21, no. 3, pp. 404–413, 2012.

[17] X. S. Zhou and T. S. Huang, "Relevance feedback in image retrieval: a comprehensive review," *Multimedia Systems*, vol. 8, no. 6, pp. 536–544, 2003.