Identifying Users' Interest Similarity Based On Clustering Hot Vertices in Social Networks

Tianchi Mo, Hongxiao Fei, Li Kuang, Qifei Qin
School of Software
Central South University (CSU)
Changsha, China
{motianchi, hxfei, kuangli, qinqifei}@csu.edu.cn

***Abstract*—Identifying users' similarity is a very important researching point because its result can be applied to many application systems. In social networks, the user circles are built not only based on their relationships in real-life, but also on common interests. Some existing approaches cannot fully capture users' similarity from the perspective of their common interests, while some other approaches are too time-consuming or space-consuming. In this paper, we propose a method of identifying users' interest similarity based on clustering Hot Vertices (HotV). A hot vertex in a social network is an account which has a large number of fans. The approach extracts users' common interests by mining and clustering the hot vertices that the two users are following simultaneously. Both the experiment and theoretical analysis have proved that the proposed approach makes a significant improvement on the precision of similarity measuring with a relatively low time and space complexity.**

Keywords-Social Network;Users' Interests Similarity; Clustering Analysis;Collaborative Filtering.

I. INTRODUCTION

With the advent of the era of Web 2.0, virtual social platform has been highly concerned by both academia and industry [1,2,3]. Nowadays, social networks can be classified into two kinds. In directed social networks like Twitter and Sina Microblog, user *u* can build relationship to user *v* without the relationship from *v* to *u*. And in undirected social networks like Facebook, the relationship between two users must be bidirectional. Directed social networks are called "interest graph", while undirected ones are called "social graph" [4].

Identifying two users' similarity in social networks is a very important research problem since its result can be applied in many fields. The recommender systems always recommend the users who have the highest similarity with user *u* to her/him [5]. Furthermore, other application systems, like personalized search engine and products recommendation systems, often use users' similarity measuring as one step of the Collaborative Filtering (CF) algorithms based on users' characteristics and actions, which is often named as user-based CF. User-based CF can be formulated into two steps as following [6]:

(1) Calculate users' similarity;

(2) Choose some users who have the highest similarity with certain user *u*, and use their history records of actions to help *u* make his decision.

So, if the precision of users' similarity measuring can be increased, the recommender systems based on user-based CF will perform better.

The similarity algorithms used by CF are often divided into two categories, namely memory-based and model-based method [6]. The memory-based methods focus on the overlap of topological information of users, items and relationships between them. While model-based methods often measure users' similarity with external information and complex mathematical model or machine learning approaches.

In social graph, the majority of users only build relationship with the people they know in the real world or they need to communicate with. But in interest graph, it is more possible that user *u* follows user *v* because *u* is interested in *v*, or *u* and *v* share similar interests, rather than they have met in real-life. Therefore the effects of users' similarity analyzing methods are different to diverse social networks. Take the friends recommendation as an example. The memory-based methods which focus on mining users' friend lists always have very good results in social graph, because users' friend lists in this kind of networks always represent their social circle in real-life. Two people with similar social circle in real-life have high probability that they need to communicate in social graph. But the case is different in interest graph. In fact, all of the existing approaches, no matter memory-based or model-based, have some defects when applied to interest graph, including:

(1) In interest graph, there are still many links based on the relationship in real-life, therefore existing approaches based on the overlap of friend list may recommend lots of acquaintances to users rather than someone who have common interest with them.

(2) For those relationships built for interest in interest graph, memory-based approaches cannot extract, feature and classify users' interest.

(3) Most of model-based methods are too time consuming to be applied in reality. At the same time, problems like information losing, data communication problem, and data sparsity may have negative effects on these methods.

To address these problems, the paper proposes an approach based on clustering Hot Vertices (HotV) in interest graph to identify users' interest similarity. A hot vertex in an interest graph is an account which has a large number of fans. HotV finds out the hot vertices in the interest graph,

978-0-7695-5381-8/14 $31.00 © 2014 IEEE
DOI 10.1109/APSCC.2014.35

and clusters them into some classes. Each class of hot vertices represents a certain interest field, and can be regarded as a certain dimension of the interest vector. The number of vertices followed by certain user in every cluster can be transformed into the value of each dimension of the interest vector of the user. And these vectors are then be used to identify users' similarity.

The rest of the paper is organized as follows. Section 2 reviews the related work about the approaches to identifying users' similarity. Section 3 introduces the proposed HotV approach. Experiments are present in Section 4. And finally, the conclusion and the future work are given in Section 5.

II. Related Work

In this section, we will introduce existing approaches on measuring users' similarity.

Memory-based approaches often use the topological relationship in the social web. Some of them measure users' similarity through calculating the overlap of topological information. This kind of approaches supposes that the two users who are more similar in social circle will have higher similarity. The most widely used memory-based methods include Common Neighbors (CN) and Jaccard coefficient [7,8,9]. Besides, methods which focus on friend list also include many transformed form of CN [7], preferential attachment method [7,8], and Adamic-Adar index [7,8,10]. There are also some methods are path-based. Classic PageRank [11,12] algorithm and Katz algorithm [5,11,13,14] both focus on the paths between two vertices, and measure the similarity through the number of paths [11].

In interest graph, the majority problems of existing memory-based methods are there are still many links based on the relationship in real-life. These methods cannot extract, feature and classify users' interest, therefore existing approaches based on the overlap of friend list may recommend lots of acquaintances to users rather than someone who share common interest with them, and users may be not satisfied with this kind of recommendation results when they are in interest graphs. If some external information, like geographic information [15], the records of sharing resource [16], and the records of login time [15,17], are imported to feature users' interest, some new problems might also be hard to deal with, like big volume of data, the sparsity of matrixes, and so on.

There are many complex mathematical models can be used to identify users' similarity, especially the similarity in interest. TF-IDF model is one of the most classic model-based approaches [4]. There are many methods which are more complex, include tensor analysis [13,18], Random walk algorithm [4,7], matrix-based methods [13,19,20], and clustering analysis[20], communities information analysis [8], exponential random graph approach [5], Markov Process [5] Monte Carlo method [5], time series model [13,15], multiple networks mutual analysis [21], users' sharing action analysis [8,16,22], recency-based method [23], Bayesian analysis [6,8], latent semantic analysis [4,6,24], Principal Component Analysis (PCA) [25], and time-agnostic maximum entropy model [17].

Most of model-based methods are very time-consuming methods. Meanwhile, ternary tensor analysis also has the defect of losing lot of binary information [13,18] about users interest similarity. And the method which uses several webs to calculate users' similarity for each other [21] has to be faced with many problems such as huge volume of data, difficulty of different networks data communication, and serious data sparsity caused by users' difference of different webs. And for some methods based on matrix transformation or clustering [8, 20], users' interest in multiple aspects and interest drifting problem may not be featured and solved well. E.g. generally, clustering algorithms often divide users into non-overlapping clusters, but, in fact, users' interest may overlap with several social circles.

III. Identifying Users' Interest Similarity Based On Clustering Hot Vertices

To address the defects mentioned above, the paper proposes an approach to identifying users' interest similarity based on clustering Hot Vertices (HotV) in interest-based social networks. HotV is designed for interest graph targetedly. Its main idea is to capture users' interest in an interest vector described by a series of clusters of hot vertices. Firstly, hot vertices in interest-based social web are found out and clustered. Each cluster represents a certain interest field and can be regarded as a dimension of a vector. Next, the interest vector of each user is evaluated by the number of vertices followed by the user in each cluster. And finally, the interest similarity of two users is measured according to their interest vectors. We will discuss each step in detail in following sub-sections.

A. Extracting Hot Vertices

Social web can be mapped into a graph structure consist of vertices and edges. Users are regarded as vertices in graphs and users' relationship are regarded as edges. And when a graph is being analyzed, the degree (in-degree or out-degree) of its vertices is very valuable and useful [5,7,15,26].

Definition 1. In social networks especially in interest graphs, a hot vertex is a user account followed by a large quantity of other users, or an account acknowledged by some organizations. The number of fans of a hot vertex is usually bigger than a relatively large threshold. It usually partly stands for a certain field of interest, and can be used to feature user's interest.

If a certain user follows a hot vertex standing for one field, then it can infer that he/she is interested in the field which the hot vertex belong to. For instance, user *u* follows many film stars in the social network, and then there is a high probability that *u* is a film lover. So, our approach should be capable of extracting the hot vertices in the social web to describe the interest. This step can separate effective and useful topological information from users' follow lists to express users' interest. And for most users, their acquaintances in real-life will not become the hot vertices in social web, and their relationships cannot stand for users' interest. So, this kind of relationship should be discarded.

There are different ways to find hot vertices in different application environments. The simplest topological way is checking the in-degree. If the number of fans of a certain user is bigger than a predefined threshold, the user can be regarded as a hot vertex. Other methods include PageRank method mentioned by [11]. If user *u*'s PR value is bigger than the threshold, then *u* is a hot vertex. And the systems can also check whether the user *u* has the VIP mark, if *u* has, *u* is a hot vertex.

B. Clustering Hot Vertices

Definition 2. A group of hot vertices which have high similarity in social relationship or labels of fields can stand for a same interest field.

According to Definition 2, if the hot vertices are clustered temperately in some aspects, they can feature users' interest for a certain field better. The step is used to fix the defect that interest fields cannot be extracted from users' follow lists in traditional memory-based methods.

The ways to cluster hot vertices are diverse. Most of existing clustering methods can be used, such as K-Means, hierarchical methods, and density-based methods. And the attributes can be used to measure hot vertices' similarity are also variable in the different application environments. In Twitter or Sina Microblog, both of their labels of status and the keywords of their posted content can be used. And besides the external information, their fans lists which are the pure topological information can also be used to measure similarity of hot vertices. For hot vertices, the overwhelming majority of links contained in their fans list are built between themselves and their fans whom they never met in real-life. So, the fans lists can be applied to describe users' interest to hot vertices, and we need not worry about the negative effect caused by acquaintance links.

The grain size of cluster needs to be controlled manually or automatically. If the average grain size is too small, hot vertices of one filed may not be divided into the same cluster. But if the grain size is too big, it might cover up the delicate differences among users' interest in a certain field. The optimal grain size needs to be judged according to the characteristic of different applications in practice.

C. Buidling Users' Interest Vector and Measuring Interest Similarity

After above two steps, suppose we can get C clusters of hot vertices. Then, these clusters can be used to describe a C-dimensional vector space. Systems can use every user's follow list to structure interest vector.

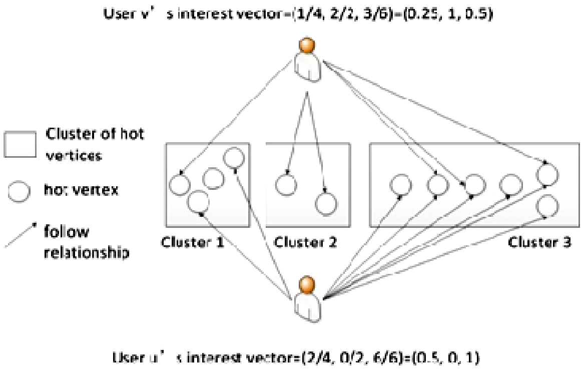
User v' s interest vector=(1/4, 2/2, 3/6)=(0.25, 1, 0.5)
Cluster of hot vertices
hot vertex
follow relationship
Cluster 1
Cluster 2
Cluster 3
User u' s interest vector=(2/4, 0/2, 6/6)=(0.5, 0, 1)

Figure 1. Building the interest vectors

The value of each dimension of a certain user's interest vector can be worked out from the number of the hot vertices that the user follows in each cluster. There is a one-to-one correspondence between a cluster and the dimension which the cluster stands for. In practice, another problem is the size of the clusters can be variable: some contain thousands of vertices, while others only have dozens. It may cause significant differences in the range of different dimensions which are represented by the clusters. To avoid the differences of users' interest in majority dimensions whose ranges are small or midsize concealed by some big dimensions, the value of each dimension should be normalized into closed interval [0, 1]. That means, to user *u*, the value of the k^{th} dimension of *u*'s interest vector is $\frac{a_k}{Num_k}$, while a_k denotes *u* follows a_k vertices in k^{th} cluster, and Num_k is the number of vertices belong to k^{th} cluster. So, user *u*'s interest vectors can be calculated by Equation (1):

$$Interest_u = < \frac{a_1}{Num_1}, \frac{a_2}{Num_2}, \ldots, \frac{a_C}{Num_C} > \qquad (1)$$

When we have the vectors, users' interest similarity can be calculated by cosine similarity as Equation (2) shows. The bigger the cosine similarity of two vectors is the more similar two users' interests are.

$$InterestSim(u, v) = \frac{Interest_u \cdot Interest_v}{|Interest_u||Interest_v|} \qquad (2)$$

This kind of topology-based vector must exist, so HotV does not have the data sparse problem. Meanwhile, multidimensional vector can quantify users' interest in different fields rather than just one field. And as for the cold start problem [4], it is easy to deal with in practical application: When a new id is registered, recommend some hot vertices to the user, and the vector can be built quickly. The proposed approach makes it possible to get enough information to describe users' interest only from the topology of the graph through extracting the hot vertices.

IV. Experiment and Evaluation

A. Design of Experiment

The dataset we used are gathered from Sina Microblog, one of the biggest social applications of interest graph type in China. The web application is very similar to Twitter. It currently has 129 million active users per month. The original dataset contains 29,268,464 rows of records. We import it into Microsoft SQL Server 2003. The format of each row of record is:

id1 id2

Each row of record means that the user whose account id is id1 follows the user whose account id is id2. The whole dataset involves of 8,327,404 users. A ***SUB-DATASET*** is extracted for the test experiment. The ***SUB-DATASET*** is built by following steps. Firstly, an initial set $\boldsymbol{S}$ is extracted. $\boldsymbol{S}$ consists of 227 ids. These ids are chosen randomly, and both the length of their follow list and length of their fan list are between 50 and 1000. Then we build user set $\boldsymbol{R}$ through $\boldsymbol{S}$, as for$\forall u \in \boldsymbol{R}$, $\exists v \in \boldsymbol{S}$ and v follows u. Next, set $\boldsymbol{Q}$ is built in the same way, as for$\forall u \in \boldsymbol{Q}$, $\exists v \in \boldsymbol{R}$ and u is in v's follow list. At last, the relationship records whose two ids (id1 and id2) are both belong to $\boldsymbol{S} \cup \boldsymbol{R} \cup \boldsymbol{Q}$ are picked out from the original dataset and form the ***SUB-DATASET. SUB-DATASET*** has 9,430,811 records in the same format as original dataset.

The test experiment is link prediction on ***SUB-DATASET***. Link prediction is often described as how to use existing information of webs to feature its development and transformation in the future. It aims to answer the question that which new links/relationships will be added into webs [11]. Friendship recommendation is the most common application of link prediction, and often used to test the users' similarity algorithms. Because we believe if two users have higher similarity in some aspects, the probability of building links between them will also be higher. The Top N mode is chosen as the recommender mode for the test. In the experiment, it means that to every user u belong to $\boldsymbol{S}$, N ($1 \le N \le 30$) users belong to $\boldsymbol{R}$ who have the highest similarity with u will be picked out. We predict that u will follow them, or we can say system recommends them to u.

In order to prove the advantage of HotV in comparison, some baselines are needed. So, existing approaches are used for prediction, too. CN method and Jaccard coefficient are chosen as the baseline methods. Equation $|a_{out} \cap b_{out}|$ for CN method and Equation $\frac{|a_{out} \cap b_{out}|}{|a_{out} \cup b_{out}|}$ for Jaccard coefficient are selected, where u_{out} denotes u's follow list and $|\boldsymbol{Set}|$ denotes the cardinality of Set. We choose follow lists instead of fan lists because "following" is a kind of more subjective action than "being followed", which can feature users' interest better. Xiang [4] also has proved that using follow lists rather than fan lists to do the prediction has the best result on the dataset of Slashdot, which is also of interest graph type.

And for the proposed approach, HotV, we set the parameters as following: if the number of users who follows user u is bigger than 500, then, u is considered as a hot vertex. We consider the threshold in the whole original dataset, not only in ***SUB-DATASET***. There are 9,406 hot vertices in ***SUB-DATASET***. Their fan lists are regarded as vectors, and they are clustered by MATLAB with K-Means function in social aspect. There are two levels of clustering: temperate and high. For temperate level, 9,406 hot vertices are clustered into 5,018 clusters; and for high level, 9,406 hot vertices are clustered into 518 clusters.

The distance measuring parameters we choose for MATLAB's K-Means function are "cosine" (cosine similarity, Cos for short) and "sqEuclidean" (Euclidean distance, Euc for short). So, we can get 4 pairs of K-Means' parameters for HotV, they are High-Cos, High-Euc, Temperate-Cos and Temperate-Euc. The results of link prediction based on 6 methods mentioned above (HotV×4, CN, and Jaccard coefficient) are worked out by JAVA console program.

B. Experiment Results

The experiment is executed as the proposal in Section 4.1, and the results are shown as Fig.2 to Fig.4. Fig.2 shows the precision of different approaches. When N (the length of recommendation list) is small, the precision of CN is higher than the precision of Jaccard coefficient. But with the growth of N, the situation is reversed. And for HotV, no matter the clustering level is "Temperate" or "High", and no matter the distance measuring parameter is "Cos" or "Euc", its results is obviously better than the baselines'. Only when N is very small and the pair of K-Means' parameters is "High-Euc", the precision is a little bit lower than CN. And with the growth of N, the "High-Euc" curve still has very clear superiority. The best pair of K-Means' parameters is "Temperate-Cos", the precision is nearly double of the precision of CN and Jaccard coefficient.

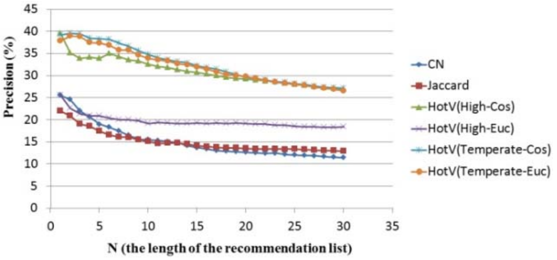

Figure 2. Precision of link prediction

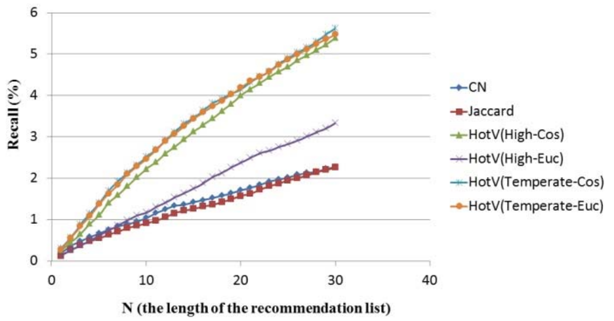

Figure 3. Recall of link prediction

And from Fig.3, it can be found that the result of recall is similar to the result of precision. No matter which pair of parameters of K-Means is chosen, the recall results of HotV are all overtly higher than that of baselines. When the grain size is "Temperate", the "Temperate-Cos" curve and the "Temperate-Euc" curve are very close to each other. With the growth of N, the two curves have the highest growth rate.

Consider Fig.2 and Fig.3 comprehensively, it can be discovered that no matter what kind of way is used to measure the distance, the prediction results based on temperately clustering (temperate grain size) are always better than that based on highly clustering (big grain size). And regardless of the clustering grain sizes, using "Cos" as distance measuring approach for K-Means is always better than using "Euc". In summary, we can get the conclusion that for the proposed approach, "Temperate-Cos" is the best pair of parameters for K-Means clustering, and "High-Euc" is the most unappealing pair to given dataset.

According to [4], coverage rate is another criterion to measure the effect of link prediction or recommend systems. If there are *a* items recommended at least once, and the number of items waiting for being recommended is *b*, then

$$\text{CoverageRate} = \frac{a}{b} \tag{3}$$

Generally, the higher the coverage rate is, the better the recommender system is, since the social web will become more active if there are many different users can be involved into the recommender system. Fig.4 shows the results of coverage rate. Except "High-Euc", HotV with other 3 pairs of parameters all have better coverage rate than traditional list-based methods (CN and Jaccard coefficient).

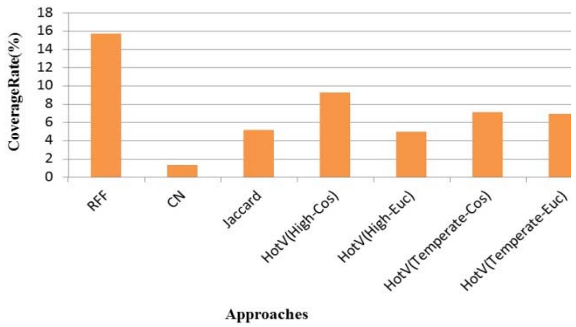

Figure 4. Coverage rate of approaches

C. Analysis and Evaluation

The proposed HotV approach performs very well in the experiment since it fixes the defects of existing memory-based methods. Existing memory-based methods just focus on the surface topological feature of the social networks, such as the overlap of friend lists, or paths between vertices. But in interest graph, this kind of methods cannot feature users' interest. Therefore the algorithms can just predict or recommend the relationship based on the similarity of users' social circle rather than the similarity of interest, and their performances are unsatisfying. HotV uses users' follow action to the hot vertices to feature users' interest, without any additional information. HotV cannot qualitatively tell what the users' interest is; however, it can build the interest vector through clustering hot vertices. Furthermore, if we have some labels of hot vertices, we can describe the interest qualitatively, too. Vectors can be used to compare different users by calculating vectors angle quantificationally and precisely. In summary, existing memory-based method cannot describe users' multiple interests through analyzing users' relationships, but HotV can. So, link prediction based on HotV has much better performance.

Meanwhile, compared with model-based methods, the advantage of HotV is that it does not need very large quantity of data. Users' relationship is relatively stable, and its data quantity is smaller than external data. Nevertheless, the fan lists or follow lists have better real-time characteristic than the labels marked by users themselves, because most of users do not change their interest labels when their interest changes, while the change can be detected from the change of hot vertices followed by them recently. If users begin to follow some hot vertices which stand for some fields they never followed before, it may imply that the users' interest is drifting.

Next step, the time complexity of HotV will be analyzed. Let us discuss the issue in a friend recommendation system. Suppose there are *m* users waiting for receiving recommendation and *n* users as the candidates for recommendation (generally, $n \approx$ the number of users of the whole application/website). Then, for existing memory-based method, for example, CN or Jaccard coefficient, the time complexity of calculating the similarity between users waiting for receiving recommendation and the candidates is $O(mn)$, and for every user waiting for receiving recommendation, the time to merge sort her/his similarity list is $O(nlog_2n)$. So, as a whole, the time complexity of collaborative filtering based on existing memory-based method for recommendation/prediction system is $O(mn)+mO(nlog_2n)=O(mn \times (1+log_2n)) \approx O(mnlog_2n)$. Considering HotV, suppose there are *h* hot vertices, they will be clustered into *C* clusters, and the vector used to describe the hot vertex is *d*-dimensional. Generally, the union set of fan lists is used to build the vector. The fans of hot vertices almost cover all users in a social application system, so might as well suppose *d=n.* In this case, the time complexity of K-Means is *O(hCd)=O(hCn)*. The time for calculating the similarity and doing the merge sort is same to CN and Jaccard coefficient, it is $O(mnlog_2n)$. So, the total time complexity of the proposed approach is $O(mnlog_2n+hCn)$. Consider that *hC<<m*, so, although the time complexity of HotV is *O(hCn)* higher than traditional memory-based methods, it wound not lengthen the running time of systems very much, like making the running time increase to $O(mn^2)$ or $O(m^2n)$. And in fact, the recommender systems do not need to cluster hot vertices every time when similarities are measured, they only need to cluster hot vertices frequently or when the system load is light. So, the complexity of HotV is almost equal to the complexity of CN and Jaccard coefficient. Taking this and the fact that the precision of HotV is twice of that of existing approaches into consideration comprehensively, we can say that HotV is very effective.

At last, the characteristics of HotV itself will be analyzed from the experiment result. Considering the precision and recall of results, it can be seen that the temperate level clustering with middle grain size is always better than high level clustering with big grain size. The cause for the phenomenon is discussed in Section 3.2, since the grain size of clustering the hot vertices should be adapted according to different application system. If the average grain size is too big, it may cover up delicate difference among users' interest in one field, then causes the decreasing of quantity of prediction/recommendation. For the Sina Microblog dataset we use, temperate level clustering is good enough. And at different clustering level, using cosine similarity as distance measuring parameter for K-Means is always better than using Euclidean distance. Because we use the fan lists to feature the hot vertices when we do K-Means clustering. It is more reasonable to regard the fans lists as vectors of hot vertices and measure the similarity with cosine formula, compared to converting the fans lists into Euclidean space and calculating the Euclidean distance between the hot vertices. Then, take the coverage rate into consideration. Consider that the proposed method has better coverage rate than CN and Jaccard coefficient in 3 conditions (High-Cos, Temperate-Cos and Temperate-Euc), there is no doubt that HotV is a better approach.

V. Conclusion and Future Work

In this paper, we propose a new approach named HotV to identifying users' similarity in interest graph. Instead of relying on external information, large quantity of data or complicated mathematical models, HotV makes it possible to build users' interest vectors and identify users' interest similarity precisely only through analyzing users' relationship, which is the simplest topological information provided by the social web itself.

The future work includes: (1) the weight of time point can be included into our model. If the time point of a user following a hot vertex is closer to current time, then, this "following" action is more capable to stand for her/his current interest, and it should have higher weight. (2) With more powerful hardware, we can employ more complicated clustering algorithms like Chameleon method [27] in HotV. And we can try to build some hybrid approaches which combine HotV with some machine learning methods to increase its precision further.

Acknowledgement

The research is supported by "National Natural Science Foundation of China" (No.61202095, NO.61073186) and "National Undergraduates' Innovation and Entrepreneurship Training Program of China" (NO. 201310533018). We are also grateful to Chen Luyi from Shanghai Jiao Tong University for providing the original dataset of Sina Microblog.

References

[1] Bellogin, A., Cantador, I., Diez, F.: An Empirical Comparison of Social, Collaborative Filtering, and Hybrid Recommenders. In: Transactions on Intelligent Systems and Technology (TIST), ACM, Vol 4 Issue 1 (2013)

[2] Konstas, I., Stathopoulos, V., Jose, J. M.: On Social Networks And Collaborative Recommendation. In: the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR' 09), New York, NY (2009)

[3] Liu, F., Lee, H. J.: Use Of Social Network Information To Enhance Collaborative Filtering Performance. J. Expert Systems with Applications. vol 37, Issue 7, pp. 4772–4778 (2010)

[4] Xiang, L.: The Practice of Recommendation System. Posts & Telecom Press, Beijing, China (2012)

[5] Zhang, C., Zhai, B. Y., Wu, M.: Link Prediction of Community in Microblog Based on Exponential Random Graph Model. In: Wireless Personal Multimedia Communications (WPMC), 2013 16th International Symposium, pp. 1-6 (2013)

[6] Wikipedia. Collaborative Filtering. http://en.wikipedia.org/wiki/Collaborative_filtering (2014)

[7] Zhou, T., Lu, L., Zhang, Y. C.: Predicting Missing Links Via Local Information. J. The European Physical Journal B, Vol 71, Issue 4, pp. 623-630 (2009)

[8] Valverde-Rebaza, J., de Andrade Lopes, A.: Structural Link Prediction Using Community Information on Twitter. In: Computational Aspects of Social Networks (CASoN), 2012 Fourth International Conference, pp. 132-137(2012)

[9] Jaccard, P.: Bulletin de la Societe Vaudoise des Sciences Naturelles. Étude comparative de la distribution florale dans une portion des Alpes et des Jura, 37, pp. 547-579 (1901)

[10] Adamic, L.A., Adar, E.: Friends and neighbors on the Web. J. Social Networks. 25, pp. 211-230 (2003)

[11] Liben-Nowell, D., Kleinberg, J.: The Link Prediction Problem for Social Networks. In: CIKM '03: The Twelfth International Conference On Information And Knowledge Management, pp. 556-559 (2003)

[12] Page L, Brin S, Motwani R, Winograd T. The Page Rank Citation Ranking: Bringing Order to the Web, http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf (1998)

[13] Acar, E., Dunlavy, D.M., Kolda, T.G.: Link Prediction On Evolving Data Using Matrix And Tensor Factorizations. In: Data Mining Workshops, 2009. ICDMW '09. IEEE International Conference, pp. 262-269 (2009)

[14] Katz, L. A New Status Index Derived From Sociometric Analysis. Psychometrika, 18(1), pp. 39-43 (1953)

[15] Scellato, S., Noulas, A., Mascolo, C.: Exploiting Place Features in Link Prediction on Location-based Social Networks. In: the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1046-1054 (2011)

[16] Go, G., Yang, J., Park, H., Han S.: Using Online Media Sharing Behavior as implicit feedback for Collaborative Filtering. In: IEEE International Conference on Social Computing / IEEE International Conference on Privacy, Security, Risk and Trust, IEEE Computer Society, pp. 439-445 (2010)

[17] Tylenda, T., Angelova, R., Bedathur, S.: Towards Time-Aware Link Prediction In Evolving Social Networks. In: SNA-KDD '09: the 3rd Workshop on Social Network Mining and Analysis. ACM(2009)

[18] Peng, J., Zeng, D., Zhao, H., Wang, F.Y.: Collaborative Filtering In Social Tagging Systems Based On Joint Item-Tag Recommendations. In: CIKM '10: the 19th ACM international conference on Information and knowledge management. ACM, pp. 809-818 (2010)

[19] Wang, J., Yin, J.: Enhancing Accuracy Of User-Based Collaborative Filtering Recommendation Algorithm In Social Network. In: System Science, Engineering Design and Manufacturing Informatization (ICSEM), 2012 3rd International Conference, pp. 142-145 (2012)

[20] Xiao, J., Zhang, Y., Jia, X., Li, T.: Measuring Similarity of Interests for Clustering Web-Users. In: Database Conference, 2001. ADC 2001. 12th Australasian, pp. 107-114 (2001)

[21] Ahmad, M.A., Borbora, Z., Srivastava, J., Contractor, N., Link Prediction Across Multiple Social Networks. In: 2010 IEEE International Conference on Data Mining Workshops, pp. 911-918 (2010)

[22] Yin, D., Hong, L., Davison, B. D.: Structural Link Analysis And Prediction In Microblogs. In: the 20th ACM International Conference on Information and Knowledge Management, ser. CIKM' 11, pp. 1163-1168 (2011)

[23] Ding, Y., Li, X., Orlowska, M.E.: Recency-Based Collaborative Filtering. In: Seventeenth Australasian Database Conference (ADC2006), pp. 99-107 (2006)

[24] Russell, M.A.: Mining the Social Web. O' Reilly Media, Inc, Sebastopol, CA (2011)

[25] Bao, Z., Zeng, Y., Tay, Y.C.: sonLP: Social Network Link Prediction by Principal Component Regression. In: the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 364-371 (2013)

[26] Barab asi, A.L., Albert, R.: Emergence of Scaling in Random Networks. J. Science, 286, pp. 509-512 (1999)

[27] Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques, 3rd Edition. Morgan Kaufmann Publishers Inc, San Mateo, CA (2012)