Integrating Meta-Path Selection with User-Preference for Top-k Relevant Search in Heterogeneous Information Networks

Shaoli Bu
School of Computer Science and Technology
Shandong University
Jinan, China
bsl89723@gmail.com

Xiaoguang Hong
School of Computer Science and Technology
Shandong University
Jinan, China
hxg@sdu.edu.cn

Zhaohui Peng
School of Computer Science and Technology
Shandong University
Jinan, China
pzh@sdu.edu.cn

Qingzhong Li
School of Computer Science and Technology
Shandong University
Jinan, China
lqz@sdu.edu.cn

***Abstract*—Relevance search in heterogeneous information networks is a basic and crucial operation which is usually used in recommendation, clustering and anomaly detection. Nowadays most existing relevance search methods focus on objects in homogeneous information networks. In this paper, we propose a method to find the top-*k* most relevant objects to a specific one in heterogeneous networks. It is a two phase process that we get the initial relevance score based on the method of pair wise random walk along given meta-paths, which is a meta-level description of the path instances in heterogeneous information networks, and then take user preference into consideration to calculate the weights combination of meta-paths and model the problem into a multi-objective linear planning problem which can be solved with the method of generic algorithm. Besides, to ensure the efficiency, we use matrix computation and selective materialization to avoid the recursive computation of pair wise random walk. What's more, we propose an effective pruning method to skip unnecessary objects computations. The experiments on IMDB and DBLP dataset show that the method can gain a better accuracy and efficiency.**

Keywords—Heterogeneous information networks; relevance search; user-preference search; graph partitioning

I. INTRODUCTION

Information networks[1] have been widely used to present the real network systems, such as social networks, bibliographic networks and so on. One problem is to find the most relevant objects for a given one in information networks which have a wide range of application in many fields of computer science, such as recommendation systems, clustering tasks, anomaly detection, and community detection and so on. A usual example is to find the similar papers for a given term or to find the most relevant movie for a user. Such methods on measuring the similarity among different types of objects in heterogeneous information networks have not been studied deeply. Most of them focus on homogeneous information networks in which nodes and links are of the same type, while it's quite necessary to do relevance search in heterogeneous information networks in which nodes and links are of multiple types.

The similarity and relevance research in information networks has come to the fore since the publishing of [1], and it has broad applications such as in the process of anomaly detection [3], search engine and recommendation systems [5]. Recently some researchers find that the measure of relevance between different-type objects is also an urgent problem and they propose *HeteSim* [2], this method could measure the relevance of objects that are of different types, such as "user-to-product", "employee-to-company" and so on. Our work integrates the meta-path selection with user preference so that we could return a more controllable result.

Traditionally, to measure the relevance among traditional numerical and categorical data types, there are *Jaccard* coefficient [7] and cosine similarity. In homogeneous information networks, based on random walks [4], [6] proposes path constrained random walks that can measure the similarity of two objects. Most recent promoted measures of the relevance are among same type of objects in heterogeneous information networks [1], such as Personalized PageRank [5], SimRank [8, 9], PathSim [10], and Scan [11]. [7] provides a measure of structural-context similarity.

Y. Sun propose to integrates meta-path selection with user guidance to cluster same type objects in [12]. It uses people's intuition that two objects are similar if they are linked by many paths in the network. It adopts probabilistic models to model the user guidance as the prior knowledge, and then uses a learning method to learn the parameters in the models and get the final result, while we take user preference as hard labeled one because the objects that follow user preference don't always in the final result.

The high computational cost of random walk and its variations is also another problem that has been considered recently. As Fujiwara Y points out in [13], many approximate approaches have the advantage of speed at the sacrifice of

978-1-4799-3775-2/14/$31.00 ©2014 IEEE

exactness. As a result, it has been difficult to utilize the approaches. Many previous works use some pruning methods such as graph partitioning [15].

We propose to integrate meta-path selection with user preference to do relevant search so with the help of user preference we could learn the most appropriate meta-path combination and their weights combination. Then the learned meta-paths with corresponding weights in turn help to find out the most relevant objects to the given specific one. For example, which community should we suggest the user join according to his interests, and which users should be grouped to the same community according to their hobbies. With the consideration of user preferences, the result is much more controllable and follows the user's guidance.

The rest of the paper is organized as follows. Section 2 introduces the data model and the problem definition. The proposed method is presented in section 3. In section 4, we do experiments to evaluate the effectiveness on real data, and then we conclude the paper in section 5.

II. PRELIMINARY

At first we define the data model and the terminologies that are going to be used in this paper. Then we describe the exact definition of the relevance search problem.

Definition 1. Heterogeneous Information Network. A heterogeneous information network is an undirected graph $G = (V, E)$ with an object type mapping function $\zeta: V \rightarrow A$ and link type mapping function $\varphi: E \rightarrow R$ and $|A| > 1$, $|R| > 1$. Each object $v \in V$ belongs to a particular type $\zeta(v) \in A$ and each link $e \in E$ belongs to a particular type $\varphi(e) \in R$.

Definition 2. Network Schema. The network schema, denoted as $T_G = (A, R)$, is a meta-level template for heterogeneous information network $G = (V, E)$ with an object type mapping function $\zeta: V \rightarrow A$ and link type mapping function $\varphi: E \rightarrow R$. Because the heterogeneous information networks in this paper are undirected, the corresponding network schemas are undirected, too.

Definition 3. Meta-path. The meta-path $P = A_0 \xrightarrow{R_1} A_1 \xrightarrow{R_2} \ldots \xrightarrow{R_l} A_l$ is a meta level description of the path instances in heterogeneous information networks defined in network schema $T_G = (A, R)$. It implies a latent relationship combination $R_1R_2 \ldots R_l$ between the type of A_0 and A_l where $A_i \in A$ and $R_i \in R$ for $i = 0,1,\ldots l$.

Definition 4. User Preference. The user preference is in the type of attribute-value pairs as $C = \{attr._1 = v_1, attr._2 = v_2, \ldots attr._q = v_q.\}$. For example, in a movie recommendation system, user preference will be like {Genre="Comedy", Year∈[2010, 2013]}, which means the user want movies of genre comedy that are distributed between the year of 2010 and 2013.

Definition 5. Top-*k* Relevance Search in Heterogeneous Networks. In heterogeneous information network $G = (V, E)$, for a given node $s \in V$ and target type T, which nodes in $V/\{v\}$ are of type T and are the top-k most relevant to s.

Definition 6. Graph Partitioning. For the given heterogeneous information network $G = (V, E)$, we partition the vertices set of V into roughly the equal parts and the number of edges connecting vertices in different parts is minimized.

In all, to implement the top-k relevance search problem, several inputs have to be specified:

1. The source object $s \in V$ that is going to be used to compute the relevance score, i.e., to find the relevant nodes to s.

2. The target type $T \in A$, which means that what kind of objects is going to be in the result set.

3. A set of user preference attributes set C.

4. A set of M meta-paths starting from type S of the source object s. We denoted them as $P_1, P_2 \ldots, P_M$.

The output of the algorithm is the set of relevant nodes of type T and their accordingly relevance score to the given source object s under the consideration of both the given M meta-paths and the user preference.

Example 1. A toy IMDB network is shown in Fig. 1. It's a typical heterogeneous information network and there are three types of objects here: *User*(U), *Movie*(M) and *Genre*(G), thus two types of links: lines between U and M represent the viewed history, whereas the links between M and G represents the genre relationship. And the question is: assume that u_1, u_2 and u_3's preferred movie genre is g_2, g_1 and g_1, respectively, which movies is the most relevant to the users and deserves to be recommended with the consideration of his/her view history and his/her interested genre.

It needs careful consideration of both the meta-path selection and the user preference during the computing procedure. For example, when we recommend a movie to user u_3 who prefers genre g_1, we can use the meta-path UMUM and preference consideration together to give m_2 recommendation. While if we just consider the UMUM meta-path, both m_2 and m_3 will be recommended without any priority. But if we integrate meta-path with user preference, i.e. g_1 here, m_2 is apparently comes before m_3.

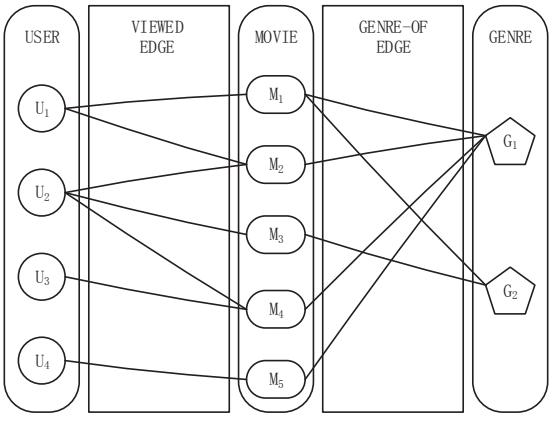
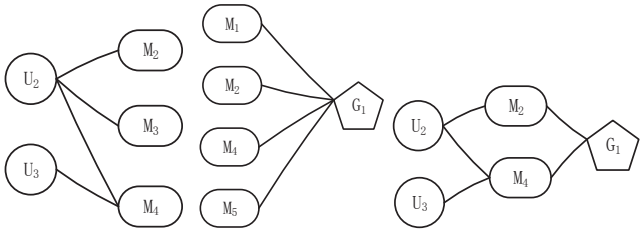


Fig. 1. A toy IMDB network



(a) UMUM (b) User Preference (c) Meta Path+User Preference

Fig. 2. Relevance Search Result under Different Conditions

As Fig. 2 interprets, in 2(a) we can easily get the relevance movies to u_3 that are m_2 and m_3 by the guidance of the UMUM meta-path which means the movies viewed by the users who have viewed the same movies as user u_3. In 2(b) it can be treated as a filter progress for a given movie genre g_2, and m_1, m_2, m_4 and m_5 satisfy. In 2(c) we combine the meta-path relevance search with user preference and give the top-2 movies that follow both UMUM meta-path selection and user preference guidance, and then we could get the priority of the recommend result.

This toy heterogeneous network shows that we can integrate user preference with meta-path based relevance search to give a better and more accurate result. For the meta-path selection, we can present it as the known input and usually it is given by domain experts. Speaking of user preference, it can be provided by users or from the users' browsing history if there are any. Usually it may not be difficult for a user to give his/her favorite genre of movie or, e.g., the preferred brand of merchandise.

III. PROPOSED METHOD

There are two naive methods when we are facing relevance search problem with the given inputs, the first one is meta-path based relevance search first and then filter the relevant search result with the given attributes and the second is of the opposite order. The drawback for the prior is that there may be no intended results in the output of the relevant search, which means we just find the relevant objects according to the given meta-paths but may not be the user preferred ones. While the drawback for the post one is that some suitable results may not follow the user guidance but still deserve to be recommended or clustered. Both of the methods will return the result with an arbitrary filtering of those target type objects that don't follow user preference. We propose a two-phase method to combine the meta-path based relevant search with user preference to get a more controllable relevant search result.

A. Framework

We tend to apply the meta-path based method to model the latent linkage of the problem we defined above, and then take the user preference into consideration to obtain the weights combination of meta-path selection. Finally we refresh the relevance score gained in the first step with the influence of weights. The result set may be different with each other every time we modify the user preference or the meta-paths selection, so we need to analyze the factors that determine a good result. First, the objects in the result set should be consistent with the given meta-paths, i.e., follow the latent link structure, we could get to the objects that have a nonzero relevance score from the source object; second, the weights combination of meta-paths should be obtained through the corporation of raw meta-paths and user preferences. As to how to get the weights combination, we intend to model it as the multi-objective linear planning problem as we can see in the third section of this chapter.

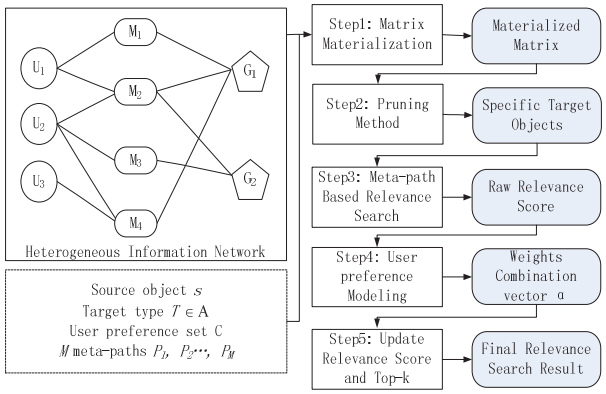


Fig. 3. Relevance Search Process

Follows are the modeling process for the above 2 aspects. Then a unified model with determined weights combination is provided. This process is demonstrated in Fig. 3.

B. Meta-path Based Relevance Score Computation

To ensure the consistency between the relevant search result and the meta-path relationship, the method we proposed is based on the given meta-paths to compute the relevance score between the source object and the target ones, i.e., the raw relevance score is accordant with the latent linkage structure that can be found out based on the given meta-paths.

The method we propose can be seen as a variant of pair wise random walk [4]. The basic idea is that both the source object and the target one start a random walk from the two ends of given meta-path, what are the chances that they will meet at the middle of the path, and the initial relevance score we get is the chances.

Following the idea we can define the meta-path based relevance measure *ReleScore* as follows:

For a given meta-path $P_l = R_1 \cdot R_2 \cdot \ldots \cdot R_l$, the relevance score *ReleScore* between the source s and target objects t is:

$$ReleScore(s,t \mid R_l \bullet R_2 \bullet \ldots \bullet R_l) = \frac{1}{\mid D(s \mid R_1) \mid \times \mid D(t \mid R_l) \mid} \times \sum_{i=1}^{D(s|R_l)} \sum_{j=1}^{D(t|R_l)} ReleScore(N_i(s \mid R_1), N_j(t \mid R_l) \mid R_2 \bullet \ldots \bullet R_{l-1}) \quad (1)$$

where $D(s|R_1)$ and $D(t|R_l)$ represent the degrees of s and t based on R_1 and R_l respectively.After the execution of *ReleScore*, we get a 2-dimentional matrix *rele* with |meta-paths| rows and |Target type objects| columns, which means that based on the given meta-path P_i, the *ReleScore* between s and t_j of Target type T is *rele*[i][j].

Next let's analyze the time and space complexity of the computing of *ReleScore*. Suppose the average size of objects is n and there are N objects of user input type. The space requirements of the *ReleScore* algorithm is just $O(N^2n^2)$ to store the final result set *rele*. Let D be the average of $|D(s|R_i)||D(t|R_j)|$ over all the pairs of (s, t) based on the meta-path R_i and R_j. for a given L-length meta-path, the time required is $O(LDN^2n^2)$.

C. Modeling the Weights Combination

Different weights combinations may lead to quite another result, so it's necessary to learn the weights combination for specific relevance search task, i.e., to consider the consistency between the meta-path based relevance search result and the user preference based one. It's clear that if the user preferred attributes are in the meta-path p_m, then the m^{th} component in the finally result matrix *ReleScore* should be larger than the one that its corresponding meta-path does not contain the user preferred attributes.

There are two reasons for a low weight of a given meta-path p_m:

- No t_j's attributes meet the user preference conditions, i.e., although we can get to t_j along the given meta-paths, but the destinations do not follow the requirements. For example, in Fig. 1, we want to recommend movies to u_3 with user preference g_2. Along the meta-path UMGM, we get m_1 , m_2 and m_4, while neither of them follows the user preference, so the given meta-path is of low quality.
- The t_j that we can get to along the given meta-paths follows the user preferences, while the length of the meta-path is too long that reduces the relevance. For example, if we are going to find the movie for u_4 and ensure that m_3 is in the final recommended results, the meta-path has to be UMGMGM or UMGMUM and the meta-path is relatively so long that the relevance score is reduced.

We get the user preference in the form of a set of attributes of the target objects and assume that all of the preferences are independent. As the objects that conform to the user preference are not necessarily in the final result, we can't regard it as the prior knowledge for the relevance search result. Thus we just treat them as hard labeled ones.

To qualify the weight of the meta-paths and take user preferences into consideration, we tend to model the condition into a multi-objective linear planning problem as: with the precondition that the sum of the weights for M meta-paths is equal to 1 and each weight is no less than 0, we have the under laid planning problem:

$$\begin{cases} \max ReleScore(s, user\ preferred\ target\ objects\ t_j), \\ \qquad j=1,2,...|user\ preferred\ objects| \\ s.t \sum_{i=1}^{m}\alpha_i = 1, \alpha_i >= 0\ for\ i = 1, 2, ...m\ where\ m = |meta-paths| \end{cases} \quad (2)$$

To maximize each *ReleScore* between the source object s and the user-preferred target objects t_j, we can deduce the above multi-objective linear planning problem to a single-objective linear planning problem as follows:

$$\begin{cases} \max \sum_{i=1}^{|meta-paths|} \sum_{j=1}^{|user\ preferred\ objects|} \alpha_i * Rele[i][j] \\ s.t \sum_{i=1}^{m}\alpha_i = 1, \alpha_i >= 0\ for\ i = 1, 2, ...m\ where\ \ m = |meta-paths| \end{cases} \quad (3)$$

Where α is a |meta-paths| dimensional vector which means the weight of corresponding meta-path. We use the generic algorithm in matlab and get the final result of α. Then based on the value of α we update the *rele*[i][j] that we calculated before, i.e., each score is multiplied by the weight parameter α_m for meta-path P_m.

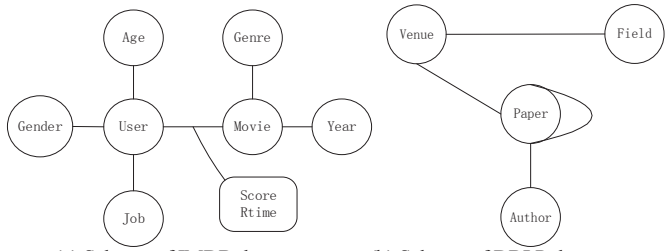
IV. EXPERIMENT

In this section, we perform the experiments on the real data set IMDB and DBLP based on the method we proposed above. The algorithm is implemented by the programming language of Java and the software of MATLAB. We evaluate the accurate rate, recall rate and the F value of our proposed method.

A. Data Set and Metrics

There are two datasets used in our experiment: IMDB and DBLP. The IMDB dataset contains 6040 users, 3952 movies of 18 movie genres and 1000209 rating records. Besides, the user has attributes of age, job type and gender, movie has the attributes of name, year, genre, and rating includes the user's score record for movie and the rating time. The DBLP dataset contains 50 venues that belong to 5 fields {DB, DM, IR, HW, MEDIA}, 57514 authors that once published their paper on such venues and 1077533 papers. So a heterogeneous information network can be formed and the corresponding schema is illustrated in Fig. 4.

For the dataset of IMDB, we sample some users who have the most viewing records and sort their rating record according to the *rtime* in the edge between user and movie, and take 75% of the records as the training set, test the relevance search result in the last 25% records. The target type is {Movie}. Besides, the user preference is obtained through the analysis of their viewing history.



(a) Schema of IMDB dataset (b) Schema of DBLP dataset

Fig. 4. Network Schema of the Two Heterogeneous Information Networks

As to the DBLP dataset, we use it to evaluate the impacts of length of meta-paths on the relevant search accurate rate as we can see in section 4.4.

We employ the accurate rate and the recall rate at *k* for top-*k* relevance search. While these two metrics are of no connection and most of the time they restrict each other so we evaluate the combination of these two indicators that is the F Value.

We focus on answering the following questions:

Q1: How accurate is the relevance search algorithm before and after graph partitioning?

Q2: How to choose the value of *k* for top-*k*?

Q3: How does the length of meta-paths influence the final result in the accuracy and recall rate?

Q4: How does the graph partitioning process influence the efficiency?

B. (Q1, Q2) Evaluation of Proposed Method with K Influence

In our experiment, we first get the raw relevance score in IMDB dataset based on the *ReleScore* algorithm in 3.1, which is a 4×3952 matrix and the 4 rows means the 4 meta-paths while each column is the corresponding movie index in the IMDB heterogeneous information network. Then according to 3.2, we model user preference into a multi-objective linear planning problem and solve it with generic algorithm.

For different users with different preferred movie genres, we will get different meta-path weights combination vector α. Using α we update the raw relevance score and get the final result. For example, for user u_1 whose user preference input is {Genre="Animation"} the best result is obtained when the meta-path weights combination vector α = [0.0037, 0.9025, 0.0226, 0.0719]. As we can see that the second component in the vector of α is the maximum one which means the second meta-path: UMGM is the strongest. This is because the user ages 4 and his most viewed and favorite movies are of genre "Animation", while most of the left users are adults and there are no much common movies they both viewed, so here it's the movie's genre that dominates the final result.

TABLE I. BASELINES OF THE EXPERIMENT

K=10	**Accurate**	**Recall**	**F Value**
UMGM	21.25%	23.46%	22.29%
UMUM	16.25%	20.93%	18.30%
UMGMGM	15.00%	18.72%	16.66%
UMGMUM	8.75%	9.24%	**89.90%**
Average	46.25%	46.30%	43.63%
Weighted	**82.25%**	**60.33%**	58.22%

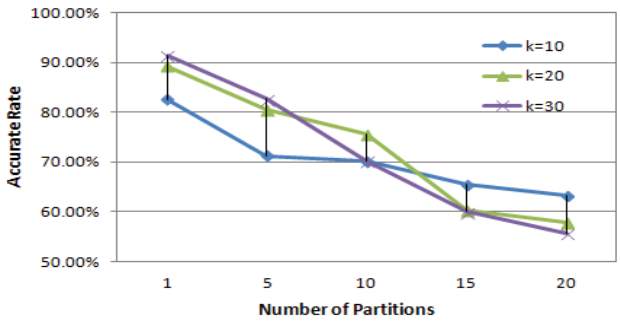


Fig. 5. Accurate Rrate (y-ais) vs. Nnumber of Partitions(x-axis)

The first 4 rows in Table 1 is obtained by the method of *HeteSim* proposed in [2] that we calculate these 4 meta-path based metrics and average the results as the Average score, while the Weighted score is obtained by our method. Comparing the Average score and that of our method, it's more easily to see that our method outperforms Average on relevant search in heterogeneous information networks. More specifically, when k = 10, our method achieves 36.00%, 14.03% and 14.59% relative improvement over Average in accurate rate, recall rate and the F value, respectively.

Besides, we compare the accuracy of relevant search before and after the process of graph partitioning. We partition the original graph into Γ parts with about the equal size using METIS [15], so that the relevant search process are only performed on the nodes that are in the same subgraph with the given source object *s* and assign 0 to those in the other sub graphs, that is, we view them as irrelevant objects. The result shows that high relevant scores don't change much however the graph is partitioned. Here we modify the number of partitions to see the variation of accuracy metric, as is shown in Fig. 5.

C. (Q3) Impacts of Length of Meta-paths

We could find some interesting issues that the length of meta-paths plays an important role in the process of deciding the relevant score. As we can see from table 1 that the longer the meta-paths are, the lower the accurate and recall rate is. For example, the accurate rate of UMGM is much higher than that of UMGMUM and UMGMGM. There are some reasons for that. Different meta-paths in heterogeneous information networks have different meanings, but they all indicate the linkage relationship between objects. Intuitively, the more steps source object goes through to meet the target one, the less relevant they are. Table 2 is the experiment result on the dataset of IMDB that displays the accurate rate based on different length of meta-paths for top-10 relevant search.

TABLE II. IMPACTS OF LENGTH OF META-PATHS ON RELEVANT SEARCH ACCURATE RATE IN IMDB.

	GM	**GMGM**	**GMGMGM**
Genre-Movie	1.0000	0.4750	0.1500
	UM	**UMUM**	**UMUMUM**
User-Movie	1.0000	0.2000	0.0750

TABLE III. TOP-10 MOST RELEVANT VENUES TO "JIAWEI HAN" BASED ON DIFFERENT META-PATHS AND (DB, DM) PREFERRED IN DBLP

Rank	Meta-path		
	APV	*APAPV*	*APAPAPV*
1	KDD	KDD	CIKM
2	ICDM	ICDE	SIGIR
3	SDM	VLDB	ICDM
4	ICDE	SIGMOD	SIGMOD
5	VLDB	ICDM	KDD
6	DOOD	EDBT	VLDB
7	SIGMOD	SDM	PAKDD
8	SSDBM	PAKDD	EDBT
9	PAKDD	WSDM	WSDM
10	EDBT	SIGIR	ICDE

And Table 3 demonstrates the top-10 most relevant conferences to Prof. Jiawei Han under 3 meta-paths that are of exactly the same base meta-path APV and his preferred conference field is in {"DM", "DB"}. The meta-paths are (APV), (APAPV) and (APAPAPV). Notice the different meanings under these 3 meta-paths. APC will find out the conferences that author published his papers in. In APAPC, author is relevant to the conferences in which his co-authors once published in. while APAPAPV relax the restriction and returns the conferences that his co-authors' co-authors once published in. As illustrated in Table 3, longer meta-paths return more global high ranked objects instead of more relevant ones.

D. (Q4) Impacts of graph partitions on the efficiency

We partition the dataset of IMDB into Γ non-overlapping sub graphs with about the same size using METIS [15]. The relevant search process is only performed on the partition that contains the source object *s* and the other nodes on different partitions are assigned 0 as their relevant scores. We evaluate the time of seconds the computation needs under 5 conditions: no partitioning, 5 sub graphs, 10 sub graphs, 15 sub graphs and 20 sub graphs.

As we can see from Fig. 6, the y-axis decreases sharply after we partition the graph, this is reasonable that our *ReleScore* is an recursive algorithm with time complexity $O(LDN^2n^2)$ where L is the average length of meta-paths, D is the average degrees of any pairs of (s, t) along the given meta-paths and n is the average size of the N objects. When the quantity of data we need to compute decreases, the N value will decrease.

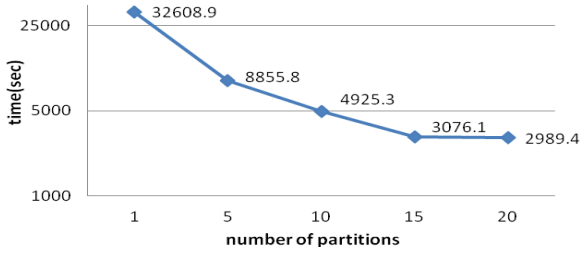


Fig. 6. Computing time (y-axis) vs. number of partitions (x-axis)

V. CONCLUSION

In this paper, we propose a method for relevance search in heterogeneous information networks. The method combines meta-path based relevance search with user preference to give a better and optimized result. With the use of meta-paths, the method could find out the latent semantic and hidden relationships in the networks, while the usage of user preference gives the user more opportunity and power to control the result and get their expected returns. Besides, we consider the relevance among not only homogeneous objects but also heterogeneous ones which is barely considered before. In the future, we will focus on the efficiency of the method. As a matter of fact, we are trying to use distributed computing methods to acculturate it.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant No.61170052) and the Natural Science Foundation of Shandong Province of China (Grant No.ZR2013FQ009).

REFERENCES

[1] Y. Sun, J. Han, X. Yan, P. Yu. Mining Knowledge from Interconnected Data: A Heterogeneous Information Network Analysis Approach[J]. Pro ceedings of the VLDB Endowment, 2012,5(12), 2022−2023

[2] C. Shi, X. Kong, P. S. Yu, S. Xie. Relevance Search in Heterogeneous Networks[C]. Extending Database Technology, 2012, 180−191

[3] J. Sun, H. Qu, D. Chakrabarti, C. Faloutsos. Relevance Search and Anomaly Detection in Bipartite Graphs[C]. SIGKDD Explorations Special Issue on Link Mining, 2005, 7(2):48–55

[4] Lovász L. Random walks on graphs: A survey[J]. Combinatorics, Paul erdos is eighty, 1993, 2(1): 1-46.

[5] G. Jeh and J. Widom. Scaling personalized web search[C]. WWW, 2003, 271−279

[6] N. Lao and W. Cohen. Fast query execution for retrieval models based on path constrained random walks[C]. Knowledge Discovery and Data Mining, 2010, 881−888

[7] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity[C]. Knowledge Discovery and Data Mining, 2002, 538−543

[8] C. Li, J. Han, G. He, X. Jin, Y. Sun, Y. Yu, and T. Wu. Fast computation of simrank for static and dynamic information networks [C]. Extending Database Technology,2010, 465−476

[9] D. Lizorkin, P. Velikhov, M. Grinev, and D. Turdakov. Accuracy estimate and optimization techniques for simrank computation [J]. PVLDB, 2008, 1(1), 422−433

[10] Y. Sun, J. Han, X. Yan, P. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks [C]. VLDB, 2011

[11] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. Scan: an structural lustering algorithm for networks [C]. Knowledge Discovery and Data Mining, 2007, 824−833

[12] 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[C]. Knowledge Discovery and Data Mining, 2012, 1348−1356

[13] Fujiwara Y, Nakatsuji M, Onizuka M, et al. Fast and exact top-k search for random walk with restart[J]. Proceedings of the VLDB Endowment, 2012, 5(5): 442-453.

[14] Tong H, Faloutsos C, Pan J Y. Fast random walk with restart and its applications[J]. ICDM, 2006, 77-79 .

[15] George Karypis, Vipin Kumar. Multilevel k-way partitioning scheme for irregular graphs[J]. Journal of Parallel and Distributed Computing, 48(1):96–129, 1998