HeteRecom: A Semantic-based Recommendation System in Heterogeneous Networks

Chuan Shi
Beijing University of Posts and Telecommunications
Beijing, China
shichuan@bupt.edu.cn

Chong Zhou
Beijing University of Posts and Telecommunications
Beijing, China
zhouchong90@gmail.com

Xiangnan Kong
University of Illinois at Chicago
IL, USA
xkong4@uic.edu

Philip S. Yu
University of Illinois at Chicago, IL, USA
King Abdulaziz University
Jeddah, Saudi Arabia
psyu@cs.uic.edu

Gang Liu
Beijing University of Posts and Telecommunications
Beijing, China
liugangofbupt@gmail.com

Bai Wang
Beijing University of Posts and Telecommunications
Beijing, China
wangbai@bupt.edu.cn

ABSTRACT

Making accurate recommendations for users has become an important function of e-commerce system with the rapid growth of WWW. Conventional recommendation systems usually recommend similar objects, which are of the same type with the query object without exploring the semantics of different similarity measures. In this paper, we organize objects in the recommendation system as a heterogeneous network. Through employing a path-based relevance measure to evaluate the relatedness between any-typed objects and capture the subtle semantic containing in each path, we implement a prototype system (called *HeteRecom*) for semantic-based recommendation. *HeteRecom* has the following unique properties: (1) It provides the semantic-based recommendation function according to the path specified by users. (2) It recommends the similar objects of the same type as well as related objects of different types. We demonstrate the effectiveness of our system with a real-world movie data set.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database applications-Data Mining

General Terms

Algorithms, Design, Experimentation

Keywords

heterogeneous information network, recommendation, similarity, semantic search

KDD'12, August 12–16, 2012, Beijing, China.

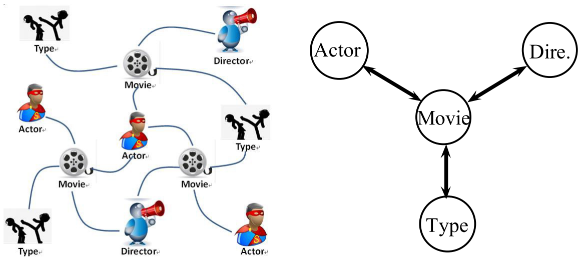
1. INTRODUCTION

With the rapid growth of WWW, we are being surrounded by a large amount of information on the web. Recommendation is an effective way to reduce the cost for finding information. It has been widely used in many e-commerce applications, such as Amazon, eBay, and Taobao.

Many recommendation methods have been proposed, which can be roughly classified into two categories: content-based filtering (CB) and collaborative filtering (CF). CB analyzes correlations between the content of the items and the user's preferences [1]. CF analyzes the similarity between users or items [2]. These methods have been applied to recommendation systems and achieved great success. However, these recommendation systems have the following disadvantages.

- Conventional recommendation systems usually recommend similar products to users without exploring the semantics of different similarity measures. However, the similar products are often different based on similarity semantics. For example, in the movie recommendation, the similar movies based on the same actors are different from those based on the same directors. Conventional systems usually give a recommendation without considering the subtle implications of similarity semantics. The proposed system is more appealing to provide a semantic recommendation function, which will give more accurate recommendation when users know their intents.
- Conventional systems only recommend same-typed objects. However, a system may be more useful if it simultaneously recommends more related objects under different semantics. For example, when users select movies, the system not only recommends the similar movies, but also suggests some related actors and directors (note that they are not limited to the actors and directors of this movie). The user may find an interesting actor and then search the movies of the actor. The relevance recommendation will provide richer information and enhance user experience.

Nowadays, social networks consisting of different types of information become popular. Particularly, the advent of



(a) Heterogeneous network of movie data (b) Network schema

Figure 1: An example of heterogeneous information network and its schema.

the Heterogeneous Information Networks (HIN) [3] provides a new perspective to design the recommendation system. HINs are the logical networks involving multiple-typed objects and multiple-typed links denoting different relations. It is clear that HINs are ubiquitous and form a critical component of modern information infrastructure [3]. Although the bipartite network [5] has been applied to organize components of recommendation system, HIN is a more general model which contains more comprehensive relations among objects and much richer semantic information. Fig. 1(a) shows an HIN example on the movie recommendation data. The network includes the richer objects (e.g., movie, actor, director) and their relations. The network structure can be represented with the star schema as shown in Fig. 1(b). HIN has an unique property [6, 7]: the different paths connecting two objects have different meanings. For example, in Fig. 1(b), movies can be connected via "Movie-Actor-Movie" (*MAM*) path, "Movie-Type-Movie" (*MTM*) path, and so on. It is clear that the semantics underneath these paths are different. The *MAM* path means that movies have the same actors, while the *MTM* path means that movies have the same types. Here the meta path connecting two-typed objects is defined as relevance path [6]. Obviously, the distinct semantics under different relevance paths will lead to different relatedness and recommendation.

Focusing on non-personalized recommendation, this paper demonstrates a semantic recommendation system, called *HeteRecom*. Different from conventional recommendation systems, it is based on HIN. Generally, *HeteRecom* has the following unique features. (1) Semantic recommendation. The system can recommend objects of the designated type based on the relevance path specified by users. (2) Relevance recommendation. Besides the same-typed objects recommendation, the system can recommend other related objects.

The implementation of *HeteRecom* faces the following challenges. (1) Relevance measure of heterogeneous objects. In order to recommend the different-typed objects, the system needs to measure the relatedness of different-typed objects. (2) The weight learning method. It is a key issue for an integrated recommendation to automatically determine the weights of different relevance paths. (3) Efficient computing strategies. In order to provide online service, the recommendation system needs to efficiently compute the relevance measure. In order to solve these challenges, the *HeteRecom* system first applies a path-based relevance measure, which can not only effectively measure the relatedness of any-typed objects but also subtly capture the semantics containing in the relevance path. Besides, a heuristic weight learning method can automatically determine the weights of different paths. Moreover, many computing strategies are designed to handle huge graph data. This paper demonstrates the effectiveness of *HeteRecom* on the real movie data through providing online semantic and relevance recommendation services.

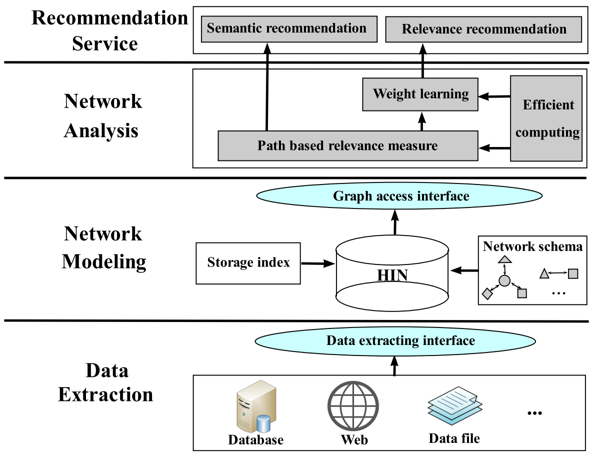


Figure 2: The architecture of *HeteRecom* system.

2. OVERVIEW OF SYSTEM

Fig. 2 shows the architecture of *HeteRecom*, which mainly consists of four components:

- Data extraction: it extracts data from different data source (e.g., database and web) to construct the network.

- Network modeling: it constructs the HIN with a given network schema. According to the structure of data, users can specify the network schema (e.g., bipartite, star or arbitrary schema) to construct the HIN database. The database provides the store and index functions of the node table and edge table of the HIN.

- Network analysis: it analyzes the HIN and provides the recommendation services. It first computes and stores the relevance matrix of object pairs by the path-based relevance measure. Based on the relevance matrix and efficient computing strategies, the system can provide the online semantic recommendation service. Through the weight learning method, it can combine the relevance information from different semantic paths and provide online relevance recommendation service.

- Recommendation service: it provides the succinct and friendly interface of recommendation services.

3. IMPLEMENTATION OF SYSTEM

It is challenging in many ways to implement these components. First, it is difficult to measure the relatedness of any-typed objects in a HIN. Second, It is not easy to combine those recommendation information on different semantic paths. Third, there are many challenges in the computation and storage of huge relevance matrix. In the following section, we will present the solutions to these challenges.

3.1 A Path-based Relevance Measure

This paper applied the *HeteSim* [6], a path-based relevance measure, to do semantic recommendation. The basic idea behind *HeteSim* is that similar objects are related to similar objects. The *HeteSim* is defined as follows.

Definition 1. ***HeteSim** [6]: Given a relevance path $\mathcal{P} = R_1 \circ R_2 \circ \cdots \circ R_l$, HeteSim between two objects s and t ($s \in R_1.S$ and $t \in R_l.T$) is:*

$$HeteSim(s,t|R_1 \circ R_2 \circ \cdots \circ R_l) = \frac{1}{|O(s|R_1)||I(t|R_l)|} \sum_{i=1}^{|O(s|R_1)|} \sum_{j=1}^{|I(t|R_l)|} HeteSim(O_i(s|R_1), I_j(t|R_l)|R_2 \circ \cdots \circ R_{l-1}) \tag{1}$$

where $O(s|R_1)$ is the out-neighbors of s based on relation R_1, $I(t|R_l)$ is the in-neighbors of t based on relation R_l, and $R.S$ ($R.T$) represents the source (target) object of relation R, respectively.

Essentially, $HeteSim(s,t|\mathcal{P})$ is a pair-wise random walk based measure, which evaluates how likely s and t will meet at the same node when s follows along the path and t goes against the path. The path implies the semantic information and $HeteSim$ evaluates the relatedness of any-typed object pairs according to the given path. The $HeteSim$ measure has shown its potential in object profiling, experts finding, and relevance search. The detailed information can be seen in [6].

Since relevance paths embody different semantics, users can specify the path according to their intents. The semantic recommendation calculates the relevance matrix with $HeteSim$ and recommends the top k objects.

3.2 Weight Learning Method

There are many relevance paths connecting the query object and related objects, so the relevance recommendation should comprehensively consider the relevance measures based on all relevance paths. It can be depicted as follows.

$$Sim(A,B) = \sum_{i=1}^{N} w_i * HeteSim(A,B|\mathcal{P}_i) \tag{2}$$

where N is the number of relevance paths, $\mathcal{P}_i$ is a relevance path connecting the object types A and B, w_i is the weight of path $\mathcal{P}_i$. Although there can be infinite relevance paths connecting two objects, we only need to consider those short paths, since the long paths are usually less important [7].

The next question is how to determine the weight w_i. The supervised learning [4] can be used to estimate these parameters. However, it is impractical for an online system: (1) It is time-consuming, even impractical, to learn these parameters on an online system. (2) It is very labor intensive and subjective work to label those learning instances. Here we propose a heuristic weight learning method.

The importance (I) of a path $\mathcal{P} = R_1 \circ R_2 \circ \cdots \circ R_l$ is determined by its strength (S) and length (l). Obviously, the path strength is decided by the strength of relations constructing the path, which can be defined as follows.

$$S(\mathcal{P}) = \prod_{i=1}^{l} S(R_i) \tag{3}$$

The strength of a relation $A \xrightarrow{R} B$ is related to the degree of A and B based on R. Intuitively, if the mutual connective links between A and B are smaller, they are more important to each other, so their relation strength is stronger. For example, the relation strength between movie and director (MD) is stronger than that between movie and type (MT). So we can define the relation strength as follows.

$$S(R) = (O(A|R)I(B|R))^{-\alpha} (\alpha \in [0,1]) \tag{4}$$

where $O(A|R)$ is the average out-degree of type A and $I(B|R)$ is the average in-degree of type B based on relation R.

The importance (I) of the path $\mathcal{P}$ is positively correlative to the path strength (S) and negatively correlative to the path length (l). Here we define it as follows.

$$I(\mathcal{P}) = f(S,l) = e^{S-l} \tag{5}$$

For multiple paths ($\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_N$), the weight ($w_i$) of path $\mathcal{P}_i$ is

$$w_i = \frac{I_i}{\sum_{i=1}^{N} I_i} \tag{6}$$

In $HeteRecom$, we consider all relevance paths whose length is smaller than a threshold Len. The relevance recommendation combines the relevance measure results of all these paths with the weight learning method and makes an integrated recommendation.

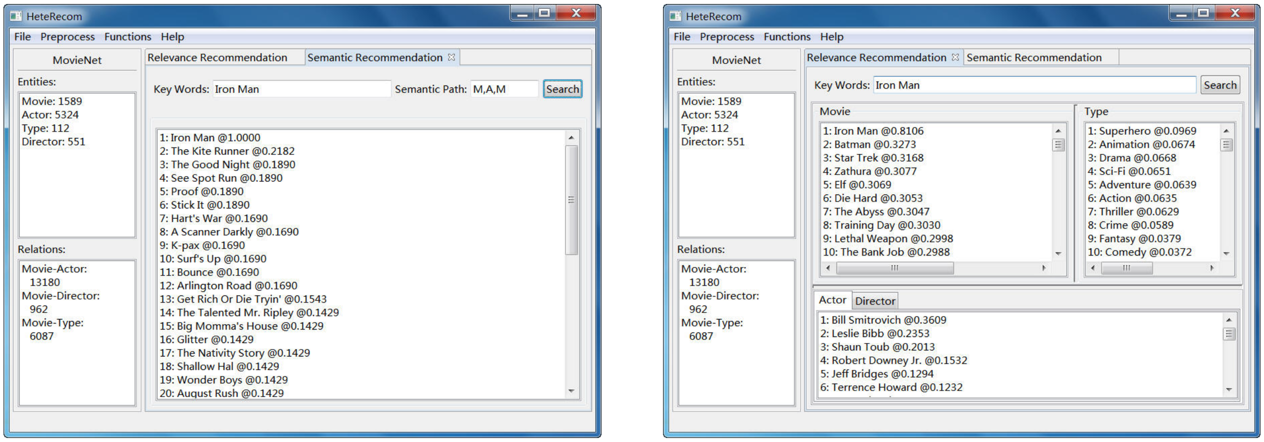
3.3 Efficient Computing Strategies

As an online recommendation system, *HeteRecom* needs to do a real-time recommendation for user's query. However, a HIN is usually huge and the computation of *HeteSim* is time-consuming. So the system employed many efficient computing strategies. Three basic strategies are depicted as follows.

Off-line computation. The primary strategy is to compute relevance matrix off-line and make recommendations online. For frequently-used relevance paths, the relevance matrix $HeteSim(A,B|\mathcal{P})$ can be calculated ahead of time. The online recommendation on $HeteSim(a,B|\mathcal{P})$ will be very fast, since it only needs to locate the position in the matrix.

Fast matrix multiplications. The most time-consuming component in the system is the matrix multiplications in *HeteSim*. There are many frequent patterns in relevance paths. Since the matrix multiplications satisfy the associative law, we can precede to compute the product of frequent patterns iteratively. Moreover, those frequent patterns only need to be computed once. For example, we only need to compute the frequent pattern AMA once for the symmetric path $AMAMA$. Since the short pattern is more frequent, we only find the most frequent relation pair in each iteration.

Matrix sparsification. The relevance matrix often becomes denser along the matrix multiplications [4]. The dense matrix may cause two difficulties. (1) It is time and space expensive to do matrix multiplications. (2) It costs a lot of time and huge memory to load and search these dense relevance matrix. As a consequence, we need to sparsify the reachable probability matrix along the matrix multiplications without much loss of accuracy. The basic idea is to truncate those less important nodes whose relevance value is smaller than a threshold ε. The static threshold [4] is not suitable, since it may truncate some important nodes



Entities:
Movie: 1589
Actor: 5324
Type: 112
Director: 551

Relations:
Movie-Actor:
13180
Movie-Director:
962
Movie-Type:
6087

Key Words: Iron Man Semantic Path: M,A,M

1: Iron Man @1.0000
2: The Kite Runner @0.2182
3: The Good Night @0.1890
4: See Spot Run @0.1890
5: Proof @0.1890
6: Stick It @0.1890
7: Hart's War @0.1690
8: A Scanner Darkly @0.1690
9: K-pax @0.1690
10: Surf's Up @0.1690
11: Bounce @0.1690
12: Arlington Road @0.1690
13: Get Rich Or Die Tryin' @0.1543
14: The Talented Mr. Ripley @0.1429
15: Big Momma's House @0.1429
16: Glitter @0.1429
17: The Nativity Story @0.1429
18: Shallow Hal @0.1429
19: Wonder Boys @0.1429
20: August Rush @0.1429

(a) Semantic recommendation based on *MAM* path

Entities:
Movie: 1589
Actor: 5324
Type: 112
Director: 551

Relations:
Movie-Actor:
13180
Movie-Director:
962
Movie-Type:
6087

Key Words: Iron Man

Movie
1: Iron Man @0.8106
2: Batman @0.3273
3: Star Trek @0.3168
4: Zathura @0.3077
5: Elf @0.3069
6: Die Hard @0.3053
7: The Abyss @0.3047
8: Training Day @0.3030
9: Lethal Weapon @0.2998
10: The Bank Job @0.2988

Type
1: Superhero @0.0969
2: Animation @0.0674
3: Drama @0.0668
4: Sci-Fi @0.0651
5: Adventure @0.0639
6: Action @0.0635
7: Thriller @0.0629
8: Crime @0.0589
9: Fantasy @0.0379
10: Comedy @0.0372

Actor Director
1: Bill Smitrovich @0.3609
2: Leslie Bibb @0.2353
3: Shaun Toub @0.2013
4: Robert Downey Jr. @0.1532
5: Jeff Bridges @0.1294
6: Terrence Howard @0.1232

(b) Relevance recommendation

Figure 3: The *HeteRecom* prototype system.

with small relevance values and keep those unimportant nodes with large relevance values. Since we usually pay close attention to the top k recommendation, we set the threshold ε as the top k relevance value of the matrix. The k is dynamically adjusted as follows.

$$k = \begin{cases} L & \text{if } L \leq W \\ \lfloor (L-W)^{\beta} \rfloor + W(\beta \in [0,1]) & \text{others} \end{cases}$$

where L is the vector length. W is the threshold which determines the size of non-zero elements. The larger W or β may lead to the denser matrix with less loss. In order to quickly determine the top k relevance value, it is approximately computed with the sample data from the raw matrix.

4. DEMONSTRATION

We showcase the *HeteRecom* prototype system using IMDB movie data as the example application. The IMDB movie data was downloaded from The Internet Movie Database [1]. The IMDB movie data collects 1591 movies before 2010. The related objects include actors, directors and types, which are organized as a star schema shown in Fig. 1(b).

Fig. 3 demonstrates the interface of the *HeteRecom* system, which is developed with Java. The left part of interface shows the basic information of the data set. The right part shows the recommendation results. In the semantic recommendation, users specify the key words and semantic path, the recommendation results will be exhibited in the panel. Fig. 3(a) shows the movies with the same actors of “Iron Man” by specified the “MAM” path. The *HeteRecom* can make many recommendations that conventional systems cannot do. For example, recommending the movies that have the same style with the movies of “Arnold Schwarzenegger” can be done by the path *AMTM*. In the relevance recommendation, the system can simultaneously recommend different-typed objects. Fig. 3(b) shows the recommendation results of the movie “Iron Man”, which includes the similar movies and related actors, directors and types. We can make many interesting recommendations on *HeteRecom*. For example, if we want to know the information about the action movie, we can search “action”. The system will recommend related action movies, actors and directors.

Note that this is an ongoing project. We will provide web service on internet and a more friendly interface will be implemented with visualization techniques. Moreover, *HeteRecom* is a general tool to do recommendation on HIN. Other recommendation tasks can be easily loaded into this system through extracting the HIN from the raw data. In addition, user preference can also be integrated into the *HeteRecom* system.

5. CONCLUSION

This paper studied the recommendation problem from the heterogeneous network angle and designed a novel recommendation system: *HeteRecom*. The *HeteRecom* system has two unique properties: semantic recommendation and relevance recommendation. The *HeteRecom* demonstrates its effectiveness on the real-world movie data set.

Acknowledgments.

It is supported by the National Natural Science Foundation of China (No. 60905025, 61074128, 61035003). It is also supported by the Fundamental Research Funds for the Central Universities.

6. REFERENCES

[1] M. Balabanovic and Y. Shoham. Content-based collaborative recommendation. *Commun. ACM*, 40(3):66–72, 1997.

[2] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI*, pages 43–52, 1998.

[3] J. Han. Mining heterogeneous information networks by exploring the power of links. In *DS*, pages 13–30, 2009.

[4] N. Lao and W. Cohen. Fast query execution for retrieval models based on path constrained random walks. In *KDD*, pages 881–888, 2010.

[5] M. Shang, L. Lu, Y. Zhang, and T. Zhou. Empirical analysis of web-based user-object bipartite networks. In *EPL 90 (0120) 48006*, 2010.

[6] C. Shi, X. Kong, P. S. Yu, and S. Xie. Relevance search in heterogeneous networks. In *EDBT*, 2012.

[7] Y. Sun, J. Han, X. Yan, P. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *VLDB*, pages 992–1003, 2011.

[1] www.imdb.com/