Relational Distance-Based Collaborative Filtering

Wei Zhang
Digital China Postdoctoral Research Workstation of Haidian Park of Beijing Zhongguancun Science Park
Beijing, China
v.cheuang@gmail.com

ABSTRACT
In this paper, we present a novel hybrid recommender system called *RelationalCF*, which integrate content and demographic information into a collaborative filtering framework by using relational distance computation approaches without the effort of form transformation and feature construction. Our experiments suggest that the effective combination of various kinds of information based on relational distance approaches provides improved accurate recommendations than other approaches.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval - Information Filtering.

General Terms
Design, Algorithms, Human Factors

Keywords
Hybrid recommender system, relational distance, collaborative filtering

1. INTRODUCTION
Recommender systems help users to deal with information overload and guide them in a personalized way to interesting or useful objects in a large space of possible options. One of the most difficult challenges for these systems is predicting a rating, which indicates how a particular user liked a particular object. Recommender systems are usually classified into the following three categories according to the sources of data on which recommendation is based. In collaborative filtering systems, the typical sources of data in a collaborative filtering system consist of a vector of items and their ratings, continuously augmented as the user interacts with the system over time. In content-based recommender systems, the objects of interest are defined by their associated features and the user will be recommended objects similar to the ones the user preferred in the past. In demographic recommender systems, the systems aim to categorize the user based on personal attributes and make recommendations based on demographic classes.

To avoid certain limitations of each pure category recommender systems, many researchers have explored hybrid recommender systems. However, the additional effort of data form transformation and feature construction is needed in most of the hybrid recommender systems. Furthermore, few of them simultaneously consider the collaborative, content and demographic information [1]. In this paper, we present a novel hybrid recommender system called *RelationalCF*, which integrate content and demographic information into a collaborative filtering framework by using relational distance approaches without the effort of form transformation and feature construction. Our experiments suggest that the effective combination of various kinds of information based on relational distance approaches provides improved accurate recommendations than other approaches.

SIGIR'08, July 20-24, 2008, Singapore.
ACM 978-1-60558-164-4/08/07.

2. RELATIONAL DISTANCE-BASED COLLABORATIVE FILTERING

2.1 Item-Based Collaborative Filtering Framework
RelationalCF is based on item-based collaborative filtering framework [2]. The item-based approach looks into the set of object items the target user has rated and computes how similar they are to the target object item i and then selects k most similar items $\{i_1, i_2, \ldots, i_k\}$. At the same time their corresponding similarities $\{s_{i1}, s_{i2}, \ldots, s_{ik}\}$ are also computed. Once the most similar items are found, the prediction, $P_{u,,i}$, on an item i for a user u is then computed by taking a sum of the ratings given by the user on the items similar to i. Formally, we can define the prediction $P_{u,,i}$ as follows.

$$P_{u,i} = \frac{\sum_{j=1}^{k} (s_{i,j} \times R_{u,j})}{\sum_{j=1}^{k} s_{i,j}}$$

Here, $R_{u,j}$ denote the rating given by the user u on the item j. Basically, this approach tries to capture how the active user rates the similar items. The weighted sum is scaled by the sum of the similarity terms to make sure the prediction is within the predefined range. Item-based collaborative filtering is a model-based approach and addresses the scalability challenge faced by common use-based collaborative filtering techniques.

2.2 Relational Distance-Based Similarity Computation
One critical step in the item-based collaborative filtering algorithm is to compute the similarity between items. Here, we want to calculate similarity between items using simultaneously content, demographic and collaborative information. Because most real-world data are stored in relational databases, these information are stored in multiple relations linked each other by referenced relationships. Thus, the challenge here is how to calculate similarity using information across the multiple relations. We employ relational distance measure over relational algebra [3]

to address exactly this problem. An example movie sales database R is adopted to illustrate our approach. The example database consists of four relations: relation *user(user id, age, gender, occupation, zip code)*, relation *movie(movie id, movie title, release date, video release date)*, relation *movie-genre(movie id, genre)*, and relation *rating(user id, movie id, rating)*. Relation *User* describes demographic information about the users. Relation *movie* describes information about the movies and relation *movie-genre* does genre information about the movies. Relation *rating* stores the ratings of movies given by users. Among them, the foreign key *user id* of relation *rating* references the primary key *user id* of relation *User*, the foreign key *movie id* of relation *rating* references the potential key *movie id* of relation *movie,* and the foreign key *movie id* of relation *movie-genre* references the potential key *movie id* of relation *movie.* These references model the one-to-many relationships between the corresponding relations. It is noteworthy that representing the database using just one big relation (i.e., the cartesian product of all 4 relations) can lead to meaning error when computing similarity between movies. Due to many one-to-many relationships between the corresponding relations, the big relation may have more than one row for one movie, which violates the general assumption in typical similarity computation that each movie has just one row in the relation.

In order to calculate similarity between movies, relation *movie* is considered as the main relation. Once the main relation is determined, similarity between movies can be calculated by using relational distance measure described as follows. Here, *similarity between movies= 1- relational distance between movies*. Each instance, M , of the *movie* relation will give rise to one *relational instance*, M^+ , i.e. an instance that spans the different relations in R. Given instance M, we create a relational instance M^+ that will have the same set of attributes and the same values for these attributes as M has. Furthermore each foreign key and the primary key add in M^+ one attribute of type set or of type list. The value of an attribute of type set will be the set or the list of instances with which M is associated in some relation when we follow one link defined by the key. These actually relational instances are retrieved by a simple SQL query. By recursive application of this procedure we obtain the complete description of the relational instance M^+. Specially, when computing the distance between relative demographic information of any pair of movie instances, we add in M^+ one attribute *DI* of type set, whose value of the attribute will be the set of instances with which M is associated in relation *user* when we follow the foreign key *movie id* and *user id* of relation *rating*.

The computation of the distance between two relational instances is done in a recursive manner traversing the full tree structures of the relational instances. For computational reasons the depth of recursion is controlled by a depth parameter *depth*. In order to compute distance *dist* between any two relational instances R_{ia}, R_{ib} of relation R_i with k attributes, we use the formula $dist(R_{ia}, R_{ib}) = \sqrt{\frac{\sum_k W_k diff^2(v_{ak}, v_{bk})}{\sum_k W_k}}$ where v_{ak}, v_{bk} are the values of the R_{ia}, R_{ib} for attribute A_k and W_k are weights for a given attribute. In our current experiments, W_ks are assigned to 1s. The sum runs over all standard attributes, all set attributes and all list attributes of the relation R_i. The function *diff* depends on the type of the attribute A_k on which it is applied. For standard continuous attributes *diff* is simply the normalized distance between two real values while for standard discrete attributes it is defined as a 0 - 1 distance. For attributes of type set and type list, different measures for defining distances between sets of objects or lists of objects can be found in [3]. In our current experiments, we use *RIBL* distances measure for distances between sets of objects and use *alignment-based edit* distance measure for distances between for lists of objects. In all cases the values of *diff* will always be between zero and one.

3. EXPERIMENTS

Two well-known datasets of movie rating are used in our experiments: MovieLens and Book-Crossing [4]. The evaluation metric used in our experiments was the commonly used mean absolute error. For MovieLens dataset, we extracted a subset of 500 users with more than 40 ratings. For Book-Crossing dataset, we extracted a subset of 10,000 users with more than 40 ratings. We compared *RelationalCF* to standard collaborative filtering approaches, including user-based Pearson Correlation Coefficient (User-Based), item-based approach (Item-Based) which only use collaborative information [2]and inductive learning approach (IL) which use collaborative and content information [1].The results of our experiments are shown in Table 1. Our experiments suggest that the effective combination of various kinds of information based on relational distance approaches provides improved accurate recommendations than other approaches.

Table 1. Experimental Results (Mean Absolute Error)

	MovieLens	Book-Crossing
User-Based	0.741	0.731
Item-Based	0.733	0.725
IL	0.727	0.717
RelationalCF	0.719	0.707

4. REFERENCES

[1] Basu, C., Hirsh, H., and Cohen, W. 1998. Recommendation as classification: using social and content-based information in recommendation. In *Proc. of the Fifteenth National/Tenth Conference on Artificial intelligence/innovative Applications of Artificial intelligence*. American Association for Artificial Intelligence, Menlo Park, CA, 714-720.

[2] Sarwar, B., Karypis, G., Konstan, J., and Reidl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proc. of the 10th international Conference on World Wide Web*. WWW '01. ACM, New York, NY, 285-295.

[3] Woznica, A., Kalousis, A. and Hilario, M. 2005. Distance-based learning over extended relational algebra structures. *As Late Breaking Papers.* In *Proc. of the 15th International Conference on Inductive Logic Programming*. ILP '05.

[4] Ziegler, C., McNee, S. M., Konstan, J. A., and Lausen, G. 2005. Improving recommendation lists through topic diversification. In *Proc. of the 14th international Conference on World Wide Web*. WWW '05. ACM, New York, NY, 22-32.