doi:10.2498/iti.2012.0404

Two-Way Collaborative Filtering on Semantically Enhanced Movie Ratings

Hasan Ogul, Emrah Ekmekciler
Department of Computer Engineering, Baskent University, Ankara, Turkey
E-mails: hogul@baskent.edu.tr, emrah.ekmekciler@gmail.com

Abstract. *A key step in recommendation systems is to estimate if a user would likely enjoy an item who has not considered yet. In this study, a new framework is defined to predict user ratings on new items from previously given ratings by other users. The systems has two major steps: (1) Enhancing available data based on semantic content to get a full item-user matrix, and (2) Predicting the unknown rating using an integrated feature set of "other ratings given by the same user" and "other ratings given to the same item". This allows the classifier to consider both user similarities and item similarities simultaneously. The system is shown to outperform existing methods in terms of prediction accuracy on a benchmark movie dataset.*

Keywords. Recommendation system, content-boosted collaborative filtering, movie rating, data mining.

1. Introduction

Recommendation systems have become extremely important in the last decade with the rapid increase in the size and complexity of data provided over the World Wide Web. Internet users usually desire to be fed by simplified and customized procedures to access the information they require. A key step in the success of a satisfactory recommendation system is the effective prediction of the rating that will be potentially given by a user to a specific item. In this way, the system can recommend the users the items that they likely enjoy.

Rating prediction step has been facing several computational challenges such as data sparsity, cold-start problem and scalability of methods (Resnick and Varian, 1997). The sparsity problem arises from the fact that a user can view and rate only a relatively low number of items in a repository. For example, a movie database may contain thousands of movies but a frequent user can potentially watch hundreds of them. Therefore, resulting matrix of item-user pairs becomes a sparse data structure, for which a straightforward statistical analysis usually fails. The cold-start problem is similar. A new user has no current rating on the repository. Therefore, the system cannot find an initialization to recommend items to new user. Since a large number of items and users are evaluated, developed methods also suffer from scalability.

Among various methods, content-boosted collaborative filtering (CBCF) has been accepted as a prominent way to get effective and efficient predictions (Herlocker et al., 2004; Claypool et al., 1999). Given a rating matrix, in which each row represents an item in the repository and each column represents a known user of the system, the CBCF approach first utilizes the content information to make a preliminary prediction from similar items' ratings. This fulfils the requirement by any classification algorithm such that the input must be a fixed-size feature vector without any missing value. Thus, second step in CBCF employs a classification rule to make a final prediction using a feature vector composed of a set of real or pseudo user ratings by others.

In this study, we propose a new CBCF system to predict user's rating on given movie item. More specifically, we attempt to predict the rating given by a specific user to a video in IMDb (International Movie Database). As other CBCF systems do, our method works in two phases. In the first phase, we use a recent content-based video similarity measure based on a newly-defined movie ontology to get a completely filled user-item matrix. This step considers several movie attributes to get video similarities in distinct hierarchical levels. In the second step, we utilize a novel two-way representation for each rating based on the combination of "same user's ratings on other video items" and "same video's rating by other users". This scheme provides a better representation since it enables inherent integration of user and video similarities. Then we predict the final rating using Support Vector Machines (SVMs), a widely-used powerful

machine learning algorithm. SVM has a good generalization ability to separate between different rating categories given in any dimensionality.

Based on the empirical tests on an available benchmark data set, our method could provide a better accuracy, in terms of F-measure, in comparison with previous methods and one-way classification approaches. Several other advantages are reported at the conclusion of the paper.

2. Methods

Recommendation systems usually exhibit in two kinds; content-based prediction and collaborative filtering (Burke 2002). In content-based predictions, the system aims at exploring similar items in terms of their content. This similarity search can be done either using signal content of the item, e.g. melody of a song, or meta-data provided in the repository, e.g. the type or singer of a song. The possibility of finding no such similarity between current item and the others may lead to failure of this scheme. The collaborative filtering approach, on the other hand, considers the similarity between users based on their current ratings. The assumption is that similar users would have a tendency to enjoy similar items. However, the assumption can not avail in case of sparsity of the data and cold-start of the user. The system proposed in this study exploits a hybrid approach defined by Melville et al. (2001). They argued that the use of two techniques together could overcome the limitations imposed by the nature of the data and would be resulted with more accurate predictions. In the first phase, they implemented content-boosting using a bag-of-words naïve Bayes classifier and used a simple neighborhood search for collaborative filtering. Ceylan and Birtürk (2011) recently proposed a new approach for content boosting based on a new semantic similarity measure and shown that it could perform better in enhancing item-user matrix. In our framework, we compiled their content-based boosting algorithm to feed a support vector machine classifier with a novel feature representation scheme. A brief outline of the framework is shown in Figure 1.

2.1. Content-based Boosting

To perform an effective model-based collaborative filtering, the learning classifier should not contain missing data. Since our item-user matrix is sparse, this problem arises more seriously in movie recommendation systems. One way of overcoming this problem is to make a preliminary prediction for missing ratings to create a pseudo-filled item-user matrix. In this scheme, the original ratings are kept while the missing values are replaced by predicted ratings. In order to incorporate the information of item similarities in recommendation model, this first stage is usually done by considering the contents of the items. Here, we use an ontology-based semantic similarity measure to evaluate item contents. The overall similarity between two movies is defined over taxonomy similarity, relation similarity and attributes similarity. The taxonomy is defined using provided IMDb metadata about movie cast, director, writer, language spoken in the movie, genre definitions, runtime, release date, country, color technology used and average rating given by other users (Ceylan and Birtürk, 2011). After calculating similarity, the prediction of unknown rating is done using a neighborhood-based method. The prediction starts with selecting k most similar items based on the semantic similarity index described above. The prediction function calculates the predicted rating of user i on item j by taking the average of ratings given by the user i on k most similar items. This procedure is iterated until all unrated item-user pairs are assigned to a pseudo rating so that a completely filled item-user matrix could be created.

2.2. Feature Representation

Each machine learning classifier requires that the input must be a fixed-length feature vector. In collaborative filtering techniques, the traditional approach is to vectorize each rating being trained or predicted by a set of known or boosted rating values given by other users. In this way, the similarity between user preferences is incorporated in classification systems. However, though having similar taste to some others, a user may have a diverge tendency in rating. While a user is giving a rate of five to the best movie, the highest credit of another user might be four. Therefore, rating strategy of a user should also be considered. To achieve this, we extended the feature vector of an item-user pair by combining the other users' ratings given to same item and same user's ratings given to the other items. More formally, a sample $s_{i,j}$ (rating to be predicted) is represented by a pair of *(i,j)*,

where i denotes the ith item of n items in the repository and j denotes the jth user of m users in the subscription network. While the common representation is $[s_{i,1}, s_{i,2}, \ldots, s_{i,m}]$, we represent each sample by $[s_{i,1}, s_{i,2}, \ldots, s_{i,m}, s_{1,j}, s_{2,j}, \ldots, s_{n,j}]$. This scheme can provide a two-way consideration of a rating, which is commonly used in clustering problems by the concept of biclustering.

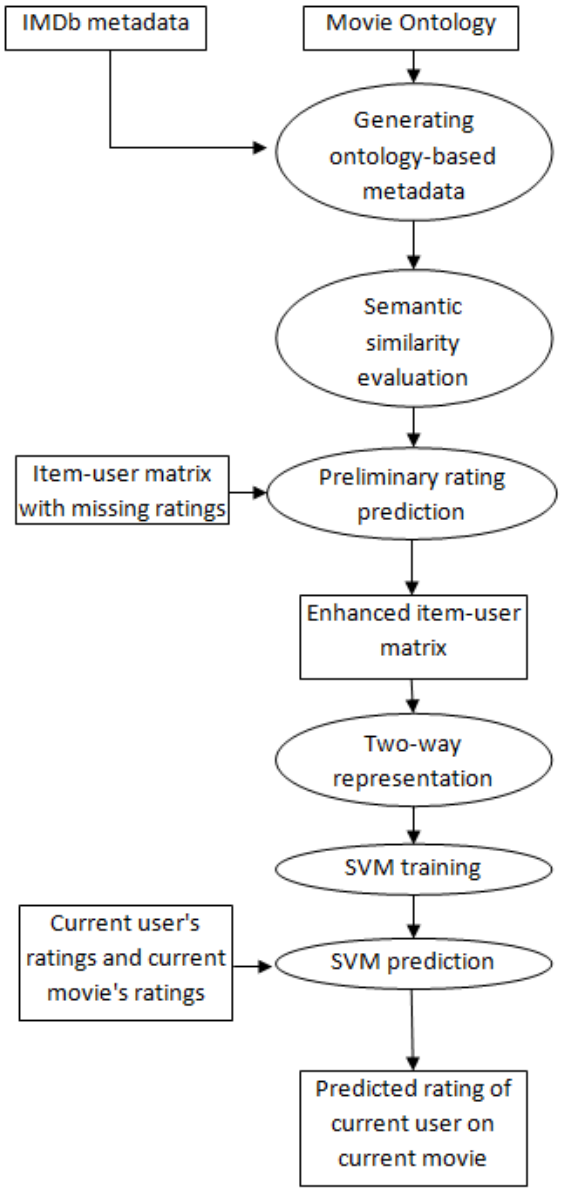

Figure 1. A brief outline of proposed system; prediction of current user's rating on current movie item.

2.3. Support Vector Machines

For training and prediction, we used a powerful machine learning method called Support Vector Machines (SVMs). SVM is a binary classifier that works based on the structural risk minimization principle. The inputs of an SVM in training phase are n-dimensional feature vectors which represent the predefined properties of the training samples. The SVM non-linearly maps its n-dimensional input space into a high dimensional feature space. In this high dimensional feature space a linear classifier is constructed. In the prediction phase, the SVM requires the feature vector corresponding to test sample prepared in the same way with the training samples. The SVM output is a discriminant score corresponding to the test sample to be classified. In a binary classification task, a positive value of this score indicates that the test sample is belonging to that class. In our application, a discriminant score higher than zero is considered to be a good rating, that it, the user will likely enjoy the item. A radial-basis-function kernel was implemented with its default parameters in LIBSVM, a publicly available SVM package (Chang and Lin, 2011) to run our experiments.

3. Results

A common setup was used to evaluate the performance of new framework and compared to several existing tools and algorithms. We include pure content-based methods, one-way colloborative filtering methods and some popular tools in our comparative study.

3.1. Dataset

We used the dataset called as MovieLens, downloaded from IMBD. The dataset has been frequently used as a benchmark suite to validate and compare collaborative filtering techniques. It consists of around 100,000 ratings from 1000 users on 1700 movies. The ratings are given between 0 and 5. In this scale, a rating that is equal to or above 3 is considered that the current movie is enjoyed by corresponding user.

3.2. Performance Evaluation

To evaluate the performance of our method, we applied 5-fold cross-validation on the disjoint test sets and their corresponding training sets. The fold partitions were selected as in original data so that a fair comparison with other methods could be conducted. We used precision, recall and F-measure to evaluate predictions. F-

measure is the weighted harmonic mean of the precision and recall, given by following equation:

$$F - measure = \frac{2 * precision * recall}{precision + recall}$$

It is used to evaluate the tradeoff between two measures, and usually considered to be a simple and effective way of comparing collaborative filtering tools.

Table 1. Comparison of methods on MovieLens data set

Method	Precision (%)	Recall (%)	F-measure (%)
MovieLens	66	74	69.8
MovieMagician (feature-based)	61	75	67.3
MovieMagician (clique-based)	74	73	73.5
MovieMagician (hybrid)	73	56	63.4
OPENMORE	75.2	73.7	74.4
ReMovender	72	78	74.9
CBCF	60	95.2	73.6
SEMCBF	63.4	92.3	75.2
SEMCBCF	63.7	93.1	75.6
Present method	**71.9**	**82.4**	**76.8**

3.3. Comparison

We compared new method with several existing tools or algorithms. Seven methods are included in comparative study. Previous results were obtained from Ceylan and Birtürk (2011). MovieLens is the tool with same name as the dataset (movielens.umn.edu). MovieMagician is a professional software tool for video playing and recommendation. OPENMORE is a content-based approach for movie recommendation (Kirmemiş and Birtürk, 2008). ReMovender is another content-based system empowered by collaborative missing data prediction (Ozbal et al., 2011). CBCF is the first hybrid approach that combines content-based methods with collaborative filtering (Melville et al. 2001). SEMCBF is the first stage of our system. It make a preliminary prediction using only the ontology-based content similarities. SEMCBCF uses one way approach to make final predictions from content-boosted item-user matrix based on the other users' rating (Ceylan and Birtürk, 2011).

Table 1 summarizes the performances of compared tools or algorithms. The results reveal that the present method can outperform the others in terms of F-measure. A higher value of F-measure indicates more accurate predictions. Although its recall is lower than other hybrid approaches, it compensates the performance by a significantly higher precision and the balance between two measures.

4. Conclusion

We present a new hybrid recommendation model which integrates a recent semantic content similarity measure and two-way classification approach. While the use of a preliminary content-based boosting step can solve cold-start and sparsity problems to a certain degree, the final system can also outperform the pure content-based approach in terms of prediction accuracy. In spite of an increase in the data dimension due to new feature representation, the system can be efficiently executed in a conventional workstation without any attempt to make it more scalable. We believe that the system can be easily adopted for other types of items and be useful for various customized e-commerce applications.

9. References

[1] Burke R, Hybrid recommender systems: survey and experiments, User Modelling and User-Adapted Interaction, vol. 12, no. 4, pp. 331–370, 2002.

[2] Ceylan U, Birturk A, Content-boosted Collaborative Filtering Using Semantic Similarity Measure, 7th International Conference on Web Information Systems and Technologies (WEBIST'11), 6-9 May 2011, Noordwijkerhout, The Netherlands.

[3] Chang C and Lin C, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 27, pp. 1-27, 2011.

[4] Claypool M, Gokhale A, Miranda T et al., Combining content-based and collaborative

filters in an online newspaper, SIGIR Workshop on Recommender Systems: Algorithms and Evaluation, Berkeley, Calif, USA, 1999.

[5] Herlocker JL, Konstan JA, Terveen LG, and Riedl JT, Evaluating collaborative filtering recommender systems, ACM Transactions on Information Systems, vol. 22, no. 1, pp. 5–53, 2004.

[6] Kirmemis O, Birturk A, A Content-Based User Model Generation and Optimization Approach for Movie Recommendation, In 6th Workshop on Intelligent Techniques for Web Personalization & Recommender Systems at the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI-08), 13-17 July 2008, Chicago, Illinois.

[7] Melville P, Mooney R, and Nagarajan R, Content-boosted collaborative filtering, In Proceedings of SIGIR-2001 Workshop on Recommender Systems, New Orleans, LA, September 2001.

[8] Ozbal G, Karaman H, Alparslan FN, A Content-Boosted Collaborative Filtering Approach for Movie Recommendation Based on Local and Global Similarity and Missing Data Prediction, The Computer Journal, vol. 54, no. 9, pp. 1535-1546, 2011.

[9] Resnick P and Varian HR, Recommender systems, Communications of the ACM, vol. 40, no. 3, pp. 56–58, 1997.

[10] Sarwar BM, Karypis G, Konstan JA and Riedl J, Item-based collaborative filtering recommendation algorithms, 10th International Conference on World Wide Web (WWW '01), pp. 285–295, May 2001.