A HYBRID RECOMMENDER SYSTEM IN E-COMMERCE USING CUSTERED CONTENT BOOSTED COLLABORATIVE FILTERING

[1]RUPAL J. SHILU, [2]DR. TEJAS P. PATALIA

[1]Research Scholar, Master in Computer Engineering(ME-CE), V.V.P. Engineering College, Rajkot-360001
[2]Head & Associate Professor, Computer Engineering Department, V.V.P. Engineering College, Rajkot-360001

[1]rupal.shilu@gmail.com,[2]pataliatejas@rediffmail.com

ABSTRACT: *Most recommender systems use Collaborative Filtering or Content-based methods to predict new items of interest for a user. While both methods have their own advantages, individually they fail to provide good recommendations in many situations. Incorporating components from both methods, a hybrid recommender system can overcome these shortcomings. In this paper, we present an elegant and effective framework for combining content and collaboration. Our approach uses a content-based predictor to enhance existing user data, and then provides personalized suggestions through collaborative filtering followed by clustering. We present experimental results that show how this approach, Clustered Content-Boosted Collaborative Filtering, performs better than a pure content-based predictor, pure collaborative filter, and a naive hybrid approach.*

KEYWORDS*: Recommender system, content-based, collaborative, hybrid recommendation, kNN, k-Mean, Correlation*

1. INTRODUCTION

Nowadays the amount of information we are retrieving have become increasingly enormous. Back in 1982, John Naisbitt observed that: “we are drowning in information but starved for knowledge." [1]. This “starvation" caused by having many ways people pour data into the Internet but not many techniques to process the data to knowledge. For example, digital libraries contain tens of thousands of journals and articles. However, it is difficult for users to pick the valuable resources they want.

One of the most successful such technologies is the Recommender system; as defined by M. Deshpande and G. Karypis:”a personalized information altering technology used to either predict whether a particular user will like a particular item (prediction problem) or to identify a set of N items that will be of interest to a certain user (top-N recommendation problem)" [2].

Over the years, various approaches for building recommender systems have been created [3]; collaborative filtering has been a very successful approach in both research and practice, and in information filtering and e-commerce applications [4]. Collaborative filtering works by creating a matrix of all items and users' preferences. In order to recommend items for the target user, similarities between him and other users are computed based on their common taste. This approach is called user-based approach. A different way to recommend items is by computing the similarities between items in the matrix. This approach is called item based approach.

2. TYPES OF RECOMMENDER SYSTEM

Recommender systems are divided according to their approach to rating estimation. The Recommender systems are classified into the following categories[5]

•**Content-based recommendations**: Based on past history[6]
•**Collaborative recommendations**: Based on similar test and preference[7].
•**Hybrid approaches**: combines more than one method[8]. (Collaborative and content-based)

Content-based methods can uniquely characterize each user, but CF still has some key advantages over them (Herlocker et al. 1999). Firstly, CF can perform in domains where there is not much content associated with items, or where the content is difficult for a computer to analyze —ideas, opinions etc. Secondly a CF system has the ability to provide serendipitous recommendations, i.e. it can recommend items that are relevant to the user, but do not contain content from the user's profile. Because of these reasons, CF systems have been used fairly successfully to build recommender systems in various domains (Goldberg et al. 1992; Resnick et al. 1994). However they suffer from two fundamental problems:

- Sparsity[9]:
  Stated simply, most users do not rate most items and hence the user-item rating matrix is typically very sparse. Therefore the probability of finding a set of users with significantly similar ratings is usually low. This is often the case when systems have a very high item-to-user ratio. This problem is also very significant when the system is in the initial stage of use.
- First-rater Problem[10]:
  An item cannot be recommended unless a user has rated it before. This problem applies to new items and also obscure items and is particularly detrimental to users with eclectic tastes. We overcome these drawbacks of CF systems by exploiting content information of the items already rated. Our basic approach uses content-based predictions to convert a sparse user ratings matrix into a full ratings matrix; and then uses CF to provide recommendations. In this paper, we present the framework for this new hybrid approach, Content-Boosted Collaborative Filtering (CBCF). We apply this framework in the domain of movie recommendation and show that our approach performs better than both pure CF and pure content-based systems.

3. ARCHITECTURE

Domain Description

We demonstrate the working of our hybrid approach in the domain of movie recommendation. The dataset contains rating data provided by each user for various movies. User ratings range from zero to five stars. Zero stars indicate extreme dislike for a movie and five stars indicate high praise. We represent the content information of every movie as a set of slots (features). Each slot is represented simply as a bag of words. The slots we use for the Each Movie dataset are: movie title, director, cast, genre, plot summary, plot keywords, user comments, external reviews, newsgroup reviews, and awards.

System Description

The general overview of our system is shown in Figure 1. The content is stored in the Movie Content Database. The EachMovie dataset also provides the user-ratings matrix, which is a matrix of users versus items, where each cell is the rating given by a user to an item. We will refer to each row of this matrix as a user ratings vector. The user-ratings matrix is very sparse, since most items have not been rated by most users. The Clustered Content-based predictor is trained on each user-ratings vector and a pseudo user-ratings vector is created. A pseudo user-ratings vector contains the user's actual ratings and content-based predictions for the unrated items. All pseudo user-ratings vectors put together form the pseudo ratings matrix, which is a full matrix. Now given an active user's ratings, predictions are made for a new item using CF on the full pseudo ratings matrix and then apply clustering for better prediction..

The following sections describe our implementation of the content-based predictor and the pure CF followed by Clustering component followed by the details of our hybrid approach.

Pure Content-based Predictor

To provide content-based predictions we treat the prediction task as a text-categorization problem. We view movie content information as text documents, and user ratings 0-5 as one of six class labels. We implemented a bag-of-words naive Bayesian text classifier [11] extended to handle a vector of bags of words; where each bag-of-words corresponds to a movie-feature (e.g. title, cast, etc.). We use the classifier to learn a user profile from a set of rated movies i.e. labeled documents. The learned profile is then used to predict the label (rating) of unrated movies. A similar approach to recommending has been used effectively in the book-recommending system LIBRA.

Pure Collaborative Filtering followed by Clustering

First implement the k-Mean[12] clustering algorithem that overcome the limitations of traditional kNN and then implement a pure collaborative filtering component that uses a neighborhood-based algorithm [13]. In neighborhood-based algorithms, a subset of users is chosen based on their similarity to the active user, and a weighted combination of their ratings is used to produce predictions for the active user. The algorithm can be summarized in the following steps[14]:

k-Mean Clustering Algorithm:

1. Initialize the value of K as the number of clusters of object to be created.

2. Generate the centroid randomly

3. Assign each object to the group that has be closest centroid

4. Update the centroid by calculating the average value of the existing data on the cluster;

$$C_i = \frac{1}{n}\sum_{j=1}^{n} d_j \quad (1)$$

Ci : centroid to-i from the cluster
n : number of object in a cluster
dj : object vector to-j

5. Repeat step 3 and 4 until the centroids no longer move (convergent). This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

6. After clustering for each category, the cluster centers were chosen to represent the category and they become the new training sets for kNN algorithm.

kNN Algorithm:

1. Weight all users with respect to similarity with the active user.
   - Similarity between users is measured as the Pearson correlation between their ratings vectors.

2. Selectn users that have the highest similarity with the active user.
   - These users form the neighborhood.

3. Compute a prediction froma weighted combination of the selected neighbors' ratings.

In step1, similarity between two users is computed using the Pearson correlation coefficient, defined below:

$$P_{a,u} = \frac{\sum_{i=1}^{m}(r_{a,i}-\bar{r_a})\times(r_{u,i}-\bar{r_u})}{\sqrt{\sum_{i=1}^{m}(r_{a,i}-\bar{r_a})^2\times\sum_{i=1}^{m}(r_{u,i}-\bar{r_u})^2}} \quad (2)$$

Where $r_{a,i}$ is the rating given to item i by user a,r_a is the mean rating given by user a and m is the total number of items. In step3, predictions are computed as the weighted average of deviations from the neighbor's mean:

$$p_{a,i} = \bar{r_a} + \frac{\sum_{u=1}^{n}(r_{u,i}-\bar{r_u})\times P_{a,u}}{\sum_{u=1}^{n} P_{a,u}} \quad (3)$$

Where $p_{a,i}$ is the prediction for the active user a for item i; $P_{a,u}$ is the similarity between users a and u; and n is the number of users in the neighborhood.

It is common for the active user to have highly correlated neighbors that are based on very few co-rated (overlapping) items. These neighbors based on a small number of overlapping items tend to be bad predictors. To devalue the correlations based on few co-rated items, we multiply the correlation by a Significance Weighting factor[15]. If two users have less than 50 co-rated items we multiply their correlation by a factor sga,u = n/50,wheren is the number of co-rated items. If the number of overlapping items is greater than 50, then we leave the correlation unchanged i.e.sga,u =1.

Clustered Content-Boosted Collaborative Filtering

In clustered content-boosted collaborative filtering, we use both methods content and collaborative followed by clustered algorithm[33]. Fig 4.1 shows the architecture of Clustered Content-Boosted Collaborative Filtering (CCBCF).

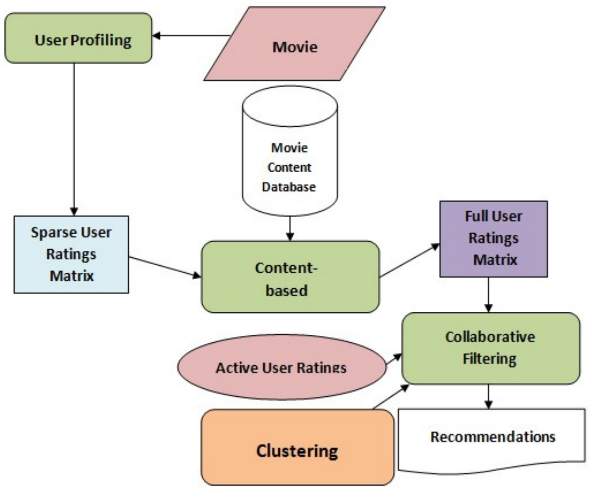

Figure 1: Architecture of CCBCF

Then we create a pseudo user-ratings vector for every user u in the database. The pseudo user-ratings vector, vu, consists of the item ratings provided by the user u, where available, and those predicted by the content-based predictor otherwise.

$$\upsilon_{u,I} = \begin{cases} r_{u,i} : \text{if user u rated item i} \\ c_{u,i} : \text{otherwise} \end{cases} \quad (4)$$

In the above equation $r_{u,i}$ denotes the actual rating provided by user u for item i, while $c_{u,i}$ is the rating predicted by the pure content-based system.

The pseudo user-ratings vectors of all users put together give the dense pseudo ratings matrix V. We now perform collaborative filtering using this dense matrix. The similarity between the active user a and another user u is computed using the Pearson correlation coefficient described in Equation 1. Instead of the original user votes, we substitute the votes provided by the pseudo user-ratings vectors v_a and v_u.

Producing Predictions Combining the above two weighting schemes, the final CBCF prediction for the active user a and item i is produced as follows:

$$p_{a,i} = \overline{v_a} + \frac{sw_a(c_{a,i} - \overline{v_a}) + \sum_{\substack{u=1 \\ u \neq a}}^{n} hw_{a,u} P_{a,u} (v_{u,i} - \bar{v}_u)}{sw_a + \sum_{\substack{u=1 \\ u \neq a}}^{n} hw_{a,u} P_{a,u}} \quad (5)$$

In the above equation $c_{a,i}$ corresponds to the pure-content predictions for the active user and item i; $v_{u,i}$ is the pseudo user-rating for a user u and item i; v_u is the mean over all items for that user sw_a, $hw_{a,u}$ and $P_{a,u}$ are as shown in Equations 4, 3 and 1 respectively; and n is the size of neighborhood. The denominator is a normalization factor that ensures all weights sum to one.

4. METHODOLOGY

We compare CCBCF to a pure content-based predictor, a CF predictor, and a hybrid approach. The hybrid approach takes the average of the ratings generated by the pure content-based predictor and the pure CF predictor. For the purposes of comparison, we used a subset of the ratings data from the Movie Lens data set (described earlier). 25 users register themselves and rate the movies as per their preferences. From each user in the test set, ratings average 11 of items were withheld. Predictions were computed for the withheld items using each of the different predictors. The quality of the various prediction algorithms were measured by comparing the predicted values for the withheld ratings to the actual ratings.

5. CONCLUSION AND FUTURE WORK

Incorporating content information into collaborative filtering can significantly improve predictions of a recommender system. In this paper, we have provided an effective way of achieving this. CCBCF elegantly exploits content within a collaborative framework. It overcomes the disadvantages of both collaborative filtering and content-based methods, by bolstering CF with content and vice versa. Further, due to the modular nature of our framework, any improvements in collaborative filtering or content-based recommending can be easily exploited to build a more powerful system. Although CCBCF performs consistently better than pure CF. The performance of our system can be boosted by using Clustered methods described earlier. In future, We will implement all traditional filtering techniques and CCBCF on dataset of movies. Then we will perform final analysis of all results of all proposed techniques.

6. REFERENCES:

[1] R. Burke. Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4):331–370, 2002.

[2] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Analysis of recommendation algorithms for e-commerce. In EC '00: Proceedings of the 2nd ACM conference on Electronic commerce, pages 158–167, New York, NY, USA, 2000. ACM Press.

[3] K. Tso and L. Schmidt-Thieme. Attribute-aware collaborative filtering. In 29th Annual Conference of the Gesellschaft fr Klassifikation (GfKl). Springer, 2005.

[4] B.M. Sarwar, G. Karypis, J. A. Konstan, and J. Reidl."Item-based collaborative filtering Recommendation algorithms". In World Wide Web, pages 285– 295, 2001.

[5] Hofmann, T. Probabilistic Latent Semantic Analysis. In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, pp. 289-296, 1999.

[6] Prem Melville, Raymond J. Mooney, Ramadass Nagarajan." Content-Boosted Collaborative Filtering for Improved Recommendations". Proceedings of the Eighteenth National Conference on Artificial Intelligence(AAAI-2002), pp.187-192,Edmonton, Canada,July2002.

[7] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7(1):76–80, January 2003.

[8] Verma, S.K., Mittal, N., Agarwal, B." Hybrid recommender system based on fuzzy clustering and collaborative filtering", Computer and Communication Technology (ICCCT), 4th International Conference,2013.

[9] Dongting S.,Cong L.,Zhigang L.” A Content enhanced approach for cold-startproblem in collaborative filtering”, 2011.

[10] Schein, A. I., A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In Proc. of the 25th Annual Intl. ACM SIGIR conf., 2002.

[11] Popescul, A., L. H. Ungar, D. M. Pennock, and S. Lawrence. Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments. In Proc. of the 17th Conf. on Uncertainty in Artificial Intelligence, Seattle, WA, 2001.

[12] Prem Melville, Raymond J. Mooney, Ramadass Nagarajan.” Content-Boosted Collaborative Filtering for Improved Recommendations”. Proceedings of the Eighteenth National Conference on Artificial Intelligence(AAAI-2002), pp.187-192, Edmonton, Canada, July 2002.

[13] Urszula Kużelewska,“Clustering Algorithms in Hybrid Recommender System on MovieLens Data”. Studies in Logic, Grammar and Rhetoric ,2014,pg 125-139.

[14] Putu Wira uana,Sesaltina Jannet D.R.M,I Ketut Gede Darma Putra,”Combination of K-Nearest Neighbor and K-Means based on Term Re-weighting for Classify Indonesian News”, International Journal of Computer Applications (0975 – 8887) Volume 50 – No.11, July 2012.

[15] K. Tso and L. Schmidt-Thieme. Attribute-aware collaborative filtering. In 29th Annual Conference of the Gesellschaft fr Klassifikation (GfKl).
Springer, 2005.