Hybrid Collaborative Filtering Based on Probabilistic Prototype

Murali Mohan.Thomurthy¹, Harada Koichi², Balakrishna.Annepu³

1. Corresponding Author: Visakhapatnam, A.P, India.
2. Professor, Department of Information Engineering, Hiroshima University, Hiroshima, Japan.
3. Assistant Prof. Noble Institute of Science and Technology. Visakhapatnam, A.P, India.

Abstract:

Collaborative filtering recommender systems recommend items by identifying other users with similar taste and use their opinions for recommendations. In this paper the concepts of Content-Boosted Collaborative Filtering algorithm is discussed and proposed for modification in the existing approach using probabilistic measures to improve the performance. The methodology using Probabilistic Prototype used for predictions have been discussed. The Formulas that were used to implement these models including Bayes conditional probability, rating, Significance Weighting Factor, Harmonic Mean Weighting and Self-Weighting and Prediction. The measured Mean Absolute Error (MAE) of the proposed model is compared with available models from literature and finally the performance analysis is done based on parameter MAE.

Keywords: Memory based, Model-based, K-NN, Item-based, Similarity, Prototype.

1. Introduction

Collaborative filtering [1] has been very successful in both research and practice, even though it has some disadvantages. For instance, it cannot recommend new items to the users and completely denies any information that could be extracted from contents of item. On the other hand, content-based [5] methods fail in providing as good recommendations as collaborative filtering does. The reason for this is that it is hard to extract really high level meaningful features. Hybrid recommendations systems are developed in the recent years as an attempt of overcome the weakness of pure content-based collaborative methods. The main idea behind hybrid recommendation techniques is that a combination of algorithms can provide more accurate recommendations than a single algorithm and disadvantages of one algorithm can be overcome by other algorithms[4] [7]. A collaborative filtering system allows increasing the quality of a recommendation [3] system by incorporating the content. Besides, when data is too sparse additional content information is a need in order to fit global probabilistic models. The work presented in explains that a method that integrates both ratings and content data enables more accurate recommendations with a richer variety than pure content-based filtering techniques. Hybrid collaborative filtering systems can exploit the content and the different similarities or dissimilarities among user preferences in explicit cases. This specific combination can be important factor for recommending truly relevant items to the user.

2. Related Work

In this work presents some ideas on how to create a content-boosted collaborative filtering system for movies. The main idea of this work is to convert a sparse user-rating matrix into a full ratings matrix using content data. Although the content-based information [6] in this case is extracted from metadata, the general idea still can be used for the specific purpose of recommendation.

1) Collaborative Filtering with Content-based for Recommendation.
2) Content-Boosted Collaborative Filtering for Recommendation.

2.1 Collaborative Filtering with Content-based for Recommendation.

Collaborative filtering recommender systems recommend items by identifying other users with similar taste and use their opinion for recommendation; whereas content-based recommender systems recommend items based on the content information of the items. It explains that a methods that integrates both ratings and content data enables more accurate recommendations with a richer variety than pure content based filtering techniques. Content-based filtering and collaborative filtering (CF) are two technologies used in recommender systems. Content-based [6] filtering systems analyze the contents of a set of items together with the ratings provided by individual users to infer which non-rated items might be of interest for a specific user. Collaborative filtering methods accumulate a database of item ratings cast by a large set of users and then use those ratings to predict user’s preferences for items. One major difficulty in designing content-based filtering systems lies in the problem of formalizing human perception and preferences. Collaborative filtering provides a powerful way to overcome these difficulties. The information on
personal preferences, tastes, and quality are all carried in either explicit or implicit user ratings.

2.2 Content-boosted Collaborative Filtering for Recommendation

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. The 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.

Content-based Predictor Algorithm:

The implementation process starts with a bag-of-words naive Bayesian text classifier to learn a user profile from a set of rated movies. Multinomial text model is adopted, in which a document is modeled as an ordered sequence of word events drawn from the same vocabulary, V. The naive Bayes assumption states that the probability of each word event is dependent on the document class. For each class $C_i$, and word, $w_k \in V$, the probabilities, $P(C_i)$ and $P(w_k | C_i)$ can be evaluated from the training data. The subsequent probability of each class given a document D is computed using Bayes rule:

$$P(C_i | D) = \frac{P(C_i) \prod_{j=1}^{d_m} P(a_{ij} | C_i)}{P(D)}$$

where $a_{ij}$ is the ith word in the document, and $|D|$ is the number of words in the document. In the implementation process, movies are represented as a vector of documents’, $d_m$, one for each slot, the probability of each word given the category and the slot, $P(w_k | C_i, S_m)$ must be estimated. The subsequent category probabilities for a film, F, is computed using:

$$P(C_i | F) = \frac{P(C_i) \prod_{m=1}^{S} \prod_{i=1}^{d_m} P(a_{mi} | C_i, S_m)}{P(F)}$$

where S is the number of slots, and $a_{mi}$ is the ith word in the mth slot.

Content-Boosted Collaborative Filtering Process:

In content-boosted collaborative filtering to generate a pseudo user-ratings vector for every user ‘u’ in the database. The pseudo user-ratings vector, $v_u$, consists of the item ratings provided by the user ‘u’ and those predicted by the content-based predictor otherwise.

$$v_u = \{ r_{ui} : \text{if user u rated item i} \}
\quad C_{ui} : \text{otherwise} \}

In the above equation $r_{ui}$ denotes the actual rating provided by user u for item i, while $C_{ui}$ 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. Collaborative filtering using this dense matrix is executed at this stage. The similarity between the active user a and another user u is computed using the Pearson correlation coefficient. Significance weighting factor is used to devalue the correlations generated based on few co-rated items to prevent bad predictors. If the number of co-rated items (n) is less than 50 then SWF is the product of their correlations. When $n \geq 50$ then the factor $S_{a,u} = 1$.

3. Proposed Model

(Methodology of Hybrid Collaborative Filtering Based on Probabilistic prototypes)

As the hybrid collaborative filtering is a combination two different techniques either one of the collaborative filtering algorithm performances of the individual components would almost certainly improve the performance of the whole system. The improvement in performance of content-based predictor or the CF algorithm, obviously, it would be able to improve total hybrid system’s predictions. A better content-based predictor would mean that the pseudo ratings matrix generated would more accurately approximate the actual full user-ratings matrix. The constant $E_{max}$ (equilibrium neighbor set) is introduced, which is the size of neighbour sets used where MAE is stable when run the modified collaborative algorithms. The test data is splitted into two different data sets to measure the $E_{max}$. Part one will be used as test data on which predictions will be measured.

4. Proposed Algorithm

Input Require: set of Items and average ratings.

Step1: A pseudo user-ratings vector for all users in the database is created by Using Harmonic Mean Weighting Factor (HMW). HMW is used to incorporate these low user-rated correlations.

$$\text{HMW} = \frac{2m_m}{m_m+m_j}$$

The HMW includes the significance weighting to obtain the hybrid correlation weight.

$$h_{a,u} = h_{m_{a,u}} + S_{a,u}$$
Step 2: Compute pseudo rating matrix V by combine the pseudo user-ratings vectors of all users.

Step 3: Compute the similarity between active user a and another user u using the Pearson Correlation coefficient.

Step 4: Compute mean-centered ratings of the best n neighbors of that user as weighted sum of the active user.

Step 5: Constant Eₜₐᵢᵢ is equilibrium neighbor set is included to calculate modified self weighting factor in the final predictions. The other neighbors are given more importance than pseudo active user. A Self Weighting(SWᵢ) factor has been incorporated in the final prediction,

$$SWᵢ = \begin{cases} \frac{(nᵢ/50)*max}{max} : & nᵢ < 50 \\ \text{Otherwise} & \end{cases}$$

where nᵢ is the number of items rated by the active user.

Step 6: Combine the above two weighting schemes to evaluate the CBCCF predictions. Combining the above two weighting schemes, the final CBCCF prediction for the active user a and item i is produced as follows:

$$pᵢ,ᵢ = \frac{SWᵢ (cᵢ,i - vᵢ) + \sum_{vᵢ \in u} hwᵢ,ᵢ pᵢ,ᵢ (vᵢ,i - vᵢ)}{SWᵢ + \sum_{vᵢ \in u} hwᵢ,ᵢ pᵢ,ᵢ}$$

Where...

- Cᵢ,ᵢ corresponds to the Content predictions for the active user and item i.
- vᵢ,ᵢ is the pseudo user-rating for a user u and item i.
- vᵢ is the mean over all items for that user.
- Swᵢ, hwᵢ,ᵢ, and Pᵢ,ᵢ are evaluated.
- n is the size of neighborhood.

5. Implementation

The implementation of the proposed model is done using JAVA. The description of implementation process is as follows:

The main java classes designed and developed to evaluate the predictions for the content-based algorithm and content-boosted algorithm are CBA5.java, NBSSimblanceRow.java, Probability.java and XYSplineRendererDemoTest.java. A segment of java code snippet and the structure of the java classes that implements the content-based collaborative filtering, collaborative filtering predictor and Content-Boosted Collaborative Filtering algorithms proposed in the system is as follows.

```
int itemsSize = 1000;
we.initialize(original, usersSize, itemsSize);
we.populateFileToList(original, fileName2, usersSize, itemsSize);

Here the List 'original' is the list which contains the original ratings of the users which will be compared with the predicted ratings. It is desired to populate the list with the ratings read from the u.data file with the mentioned Path in the code.

List test = new ArrayList();
fileName2 = "D:\Excelwork\ml-data_0\u5.test";
we.initialize(test, usersSize, itemsSize);
we.populateFileToList(test, fileName2, usersSize, itemsSize);

The List 'test' is the list which contains the test ratings of the users. Test data is the subset of original data. Using test data, it is designed to produce the user rating predictions and populating the 'test' list with values read from u5.test. Content-based predictions are generated by treating the task as a text-categorization problem. The movie data which contains the content information is considered as text documents, and user ratings given as 1-5.

FILENAME2 = "D:\Excelwork\ml-data_0\u.item";
ArrayList genre = new ArrayList();
we.initializeGenre(genre, fileName2);

The List 'genre' is the list of genre of the movies. Each movie genre is given a unique number which is used in item classification.

FILENAME2 = "D:\Excelwork\ml-data_0\u.item";
List items = new ArrayList();
we.initializeItems(items, 1682, 30);
we.populateItemsToList(items, fileName2, 1682, 30, genre);

The List 'items' is the list of all the items that presented in u.item. 1682 is number of items given, and 30 is the number of properties mentioned in the u.item file. It is designed and developed to populate the test list with values read from u.item. All the properties are embedded in a child list and the child list is added to parent list.

List docsIJ = new ArrayList();
we.initialize(docsIJ, usersSize, 5);
we.populateNoOfRatingsVsClazz(docsIJ, test, usersSize);

The List docsIJ is the list that contains the data of users which rated for first grade (rating given as 1). Similarly second grade and so on ratings given by particular user. Hence it contains number particular ratings (1-5) which was graded by each and every user. The method populateNoOfRatingsVsClazz is designed to develop the list of the ratings of all users.

List Examples = new ArrayList();
we.populateExamples(Examples, docsIJ); 'Examples' is the list of number of total ratings given by every user. It is...
designed and developed to generate MAE values for content boosted collaborative filtering.

MAE calculates the irrelevance between the recommendation value predicted by the system and the actual evaluation\[9] \[10] value rated by the user. The measurement method of evaluating the recommendation quality of recommendation system mainly includes statistical precision measurement method it includes to measure the recommendation quality.

The generated prediction values are stored in an arraylist of Examples mentioned above and tabulated in the next sections. This arraylist Example is input for the the class \textit{populatePredictContentBoostedUJ} to generate the MAE values. The arraylist contains the values of the predicted user rating is generated with the java class ‘Examples’ and actual user rating arraylist generated with the java class ‘original’. These two arraylist are the inputs for \textit{populatePredictContentBoostedUJ} java class to generate MAE values:

\begin{verbatim}
  sheetData1 = ((List)((ArrayList)test).clone());
  s3 = new ArrayList();
  Hashtable table = new Hashtable();
  for(int i=4; i<30; i=i+4)
    // s3 = we.populatePredictUJ(sheetData1, listSimblances, i);
    s3 = we.populatePredictContentBoostedUJ(test1, test2, Examples, listSimblances, i);
    double mae = we.getMAE1(test, original, s3);
    BigDecimal z1 = new BigDecimal(mae).setScale(2, BigDecimal.ROUND_HALF_UP);
    System.out.println("For neighbour size -- " + i + " MAE is " + mae);
    table.put(new Double(i), mae);
\end{verbatim}

When the user search information, the system use the search key words that the user fills out as the representation of user interest key words, store them in the user interest table, and then assign them weight value. When the user collect some information, it can conclude that the user is interested in such information, some words are taken out as the user interest keywords and store them in the user interest table, assign them weight value, use the number of times to represent the weight value of the key words.

6. Model Experimentation

The research and analysis [7] for user-based collaborative filtering system is carried out using MovieLens dataset which is available for research purpose provided by the GroupLens [2] Research Project agency at the University of Minnesota. As it is mentioned the chapter 3, the dataset consists of 100,000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies. It provides demographic data such as age, gender, and the zip code supplied by each person. The content of the information of every movie is considered as a set of slots. Each slot is represented by number of words. Further, the data has been segregated and discarded for having less than 20 ratings or in complete demographic information. A subset of the ratings data from the MovieLens data set used for the purposes of comparison. 20% of the users were randomly selected to be the test users. The data sets u1.base and u1.test through u5.base and u5.test are 80%/20% splits of the u data into training and test data. Each of u1, u2, u3, u4, and u5 has disjointed test sets for cross validation.

7. Results and Discussions

The MAE values are computed using existing content-boosted collaborative filtering (CBCF) and modified CBCF for U1.test, U2.test, U3.test, U4.test and U5.test for test dataset and tabulated in table 1 to table 5.

The Comparative analysis of these computed values are presented.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline
\textbf{Neighbor Set Size} & 4 & 8 & 12 & 16 & 20 & 24 & 28 \\
\hline
\textbf{MAE for Existing} & 0.98 & 0.91 & 0.89 & 0.86 & 0.85 & 0.85 & 0.85 \\
\textbf{MAE for Proposed Mode} & 0.82 & 0.81 & 0.81 & 0.81 & 0.81 & 0.81 & 0.81 \\
\hline
\end{tabular}
\caption{MAE for different neighbor sets for CF on U1.test dataset}
\end{table}

![Fig 1. Comparison of MAE for content-boosted collaborative filtering (CBCF) algorithm vs modified algorithm on the U1.test dataset.](image)

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline
\textbf{Neighbor Set Size} & 4 & 8 & 12 & 16 & 20 & 24 & 28 \\
\hline
\textbf{MAE for Existing} & 1.07 & 0.98 & 0.94 & 0.92 & 0.91 & 0.90 & 0.90 \\
\textbf{MAE for Proposed Mode} & 0.86 & 0.86 & 0.86 & 0.86 & 0.86 & 0.86 & 0.86 \\
\hline
\end{tabular}
\caption{MAE values CBCF on U2.test dataset}
\end{table}
The results presented in this chapter are given according to evaluation procedures with the experiments performed. Proposed user-based collaborative filtering algorithm and content-based predictor algorithm will be combined in content-boosted collaborative filtering (CBCF), the evaluated results are tabulated. Derived MEA values for different test datasets from U1.test to U5.test is related with recommendation accuracy which is computed and compared for both existing and modified methods to see which one performs better. MAE is obtained for every fold in our 5-fold cross validation experiment. Finally the total MAE was computed from the whole set of users and folds in the experiments.

8. Conclusion

Hybrid recommendations systems are developed to overcome the weakness of traditional content-based collaborative systems and collaborative filtering algorithms. The design and development of models allowed the system to increasing the quality of a recommendation system when data is too sparse additional content information is a need in order to fit global probabilistic models. This chapter presents a modified content-boosted collaborative filtering system used the MovieLens dataset which contains user ratings on movies. One of the key factors of content-boosted collaborative filtering algorithm is to convert a sparse user-rating matrix into a full ratings matrix using content data. Although
content-based information in this case is extracted from metadata, the general idea still can be used for the specific purpose of recommendation. It is thus suggested to include the content of items into a collaborative-filtering system in order to improve the quality of its predictions and to solve the cold start problem.

9. References

AUTHOR’S PROFILE
Thomurthy Murali Mohan is a Assistant Professor from Kaushik College of Engineering, affiliated to Jawaharlal Nehru Technological University. He received the B.Sc (Computer Science), M.Sc (Computer Science) in 2003 & 2005 from Andhra University. He Completed M.Tech (Computer Science & Engineering) in 2012 from Jawaharlal Nehru Technology University, Kakinada. He received MBA in 2010 from Punjab Technical University. His current research is mainly in Data Mining, and Data Warehousing. Special interests include in Robotics.

Koichi Harada is a professor of the graduate Schools of Engineering at Hiroshima University. He received the BE in 1973 from Hiroshima University, and MS and Phd in 1975 and 1978, respectively, from Tokyo Institute of Technology. His current research is mainly in the area of the computer graphics. Special interests include man-machine interface through graphics; 3D data input techniques, data conversion between 2D and 3D geometry, effective interactive usage of curved surfaces. He is a member of ACM, IPS of Japan, and IEICE of Japan

Annepu Balakrishna is a professor from Noble Institute of Engineering college, Visakhapatnam. He received MCA in 2009 from Andhra University, and M.Tech in 2013 from JNT University, Kakinada and He Ph.D in 2012 from CMJ University. His area of interestes are Software Engineering. His current research is mainly in Data Mining and Data Ware Housing. Special interestes include in Robotics.