Content-boosted Collaborative filtering approach to reduce Cold Start and Data Sparsity problems

Raja Sarath Kumar Boddu

January 6, 2020

Content-boosted Collaborative filtering approach to reduce Cold Start and Data Sparsity problems

Raja Sarath Kumar Boddu
Computer Science Department,
Lenora College of Engineering, India
iamsarathphd@gmail.com

Abstract. Recommendation systems suffer from problems related to scalability, data sparsity and cold starts, resulting in poor-quality predictions. Hybrid techniques, such as content-boosted collaborative filtering (CBCF) and/or combine collaborative filtering methods with other recommendation systems are highly essential to alleviate the drawbacks and to improve the overall prediction rate. Obviously, the combination of algorithms could make more accurate recommendations. CBCF could be used with a combination of a pure content-based predictor (pure CF) and user-based collaborative filtering (UBCF), which improves prediction quality and thus minimizes cold start and data sparsity problems. In this paper, a modified CBCF algorithm by implicitly collecting user ratings through a user-interest model has been developed. Experimental results were tabulated.

Keywords. Cold start; Content-boosted Collaborative Filtering; Correlation Similarity; Data Sparsity; Mean Absolute Error

1 Introduction

Recent changes in the World Wide Web and the enormous amount of accessible information have led to a rapid rise in the number of Internet users. This growing Internet usage limits the effectiveness of search engines and presents a challenging problem to all recommendation systems. Recommendation systems generate suggestions of items or products that might interest a collection of users [1]. Accurate and high-performance recommender systems can provide personalized services for users. Collaborative filtering models are useful for recommendations in situations where user profiles are investigated by observing a user’s transactions with a system, such as the user’s rating of a product that he has purchased [2]. Collaborative filtering algorithms must have the ability to deal with highly sparse data, to scale with an increasing numbers of users and items and make adequate recommendations, and to deal with problems such as cold start, data sparsity, scalability, synonymy, grey sheep, shilling attacks, and privacy protection. The idea behind CBCF prediction techniques is that a combination of algorithms can provide more accurate recommendations than a single algorithm [3]. The main drawbacks of CF algorithms are when a user does not rate many items and an item that can only be recommended after it has been rated by users which are commonly denoted as sparsity and cold start problems.

1.1 Cold Start-The first of these problems arises in collaborative filtering systems, where an item cannot be recommended unless a user has rated it. Collectively, these problems are referred to as the cold start problem [4].

1.2 Data sparsity-The data sparsity challenge appears in several situations. Specifically, the cold start or new item problem occurs when a new user or item has just entered the system. In such scenarios, it is difficult to find similar instances against which to form a comparison because of a lack of information [5].

2 Methodology

As hybrid collaborative filtering is a combination of two or more different techniques, an improvement in the performance of either technique might be useful to improve the overall performance. A better content-based predictor would mean that the pseudo-ratings matrix generated would more accurately approximate the actual user-ratings matrix. Hence, a user-based collaborative filtering algorithm as a CF algorithm is proposed. This has already been shown to be the best algorithm in the memory-based collaborative filtering classification [6]. In a pure content-based predictor, a naive Bayesian text classifier is used to learn a six-way classification task. It is proposed to improve content-based predictions using this approach [7]. 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, each of them by itself fails to provide suitable recommendations in many situations. By incorporating components from both methods, the resulting hybrid recommender system can overcome these shortcomings. An effective framework for combining content and collaboration has been presented. A five-fold cross-validation with the predictions of the evaluation users to properly evaluate the prototypes has been proposed. It has been proposed to split the user evaluation ratings 10 times for experimental point of view. These sets of ratings are used each time to randomly select10% of the ratings of every user and store them in a table. The other 90% of the ratings are stored in a training table.

3 Pure Content-based Predictor

The implementation process starts with a Bayesian text classifier to learn a user profile from a set of rated movies. The prediction task can be assumed to be a text categorization problem, movie content information can be taken to be text documents, and user ratings (1-5) can be seen as one of six class labels. A 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 [8] 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 $P(w_k \mid C_i)$ can be calculated from the training data. The subsequent probability of each class given a document D is computed using Bayes' rule:

$$P\left(C_t \middle| D\right) = \frac{P\left(C_t\right)}{P\left(D\right)} \prod_{t=1}^{|D|} P\left(a_t \middle| C_t\right) \tag{1}$$

Where

a_i - i^{th} Word in the document and

$|D|$ - Number of words in the document.

In the implementation process, movies are represented as vectors of documents 'd_m' one for each slot. The probability of each word given the category and the slot $P(w_k \mid C_{i,} S_m)$ must be calculated. The subsequent category probabilities, F, are computed using

$$P(C_i \mid F) = \frac{P(C_i)}{P(F)} \prod_{m=1}^{S} \prod_{i=1}^{|d_m||} P(a_{mi} \mid C_{i,} S_m) \tag{2}$$

Where

S - Number of slots and

a_{mi} - i^{th} Word in the m^{th} slot.

After evaluating and implementing the predictor, the generated prediction quality value of the MAE is 0.982.

4 User-based Collaborative Filtering (UBCF) Algorithm

The UBCF algorithm produces a recommendation list for an object user according to the views of other users. The assumptions are that if ratings for a few items by a few users are similar to one another, ratings of other items by these users will also be similar [4]. The CF recommendation system uses statistical techniques to search the nearest neighbors of the object user. Based on the item rating awarded by the nearest neighbors, it predicts the item rating awarded by the object user and produces a corresponding recommendation list.

4.1 Pearson Correlation Similarity - Similarity between two users and/or two items is measured by correlation-based similarities, such as the Pearson correlation. Pearson correlation measures the extent to which two variables linearly relate to each other. It measures the extent to which two variables linearly relate to each other. Hence, it is used to measure the similarity of various collaborative filtering algorithms. The Pearson correlation between users is

$$P_{a,u} = \frac{\sum_{i=1}^{rn}(r_{a,i} - \bar{r}_a) \times (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{rn}(r_{a,i} - \bar{r}_a)^3 \times \sum_{i=1}^{rn}(r_{u,i} - \bar{r}_u)^3}} \qquad (3)$$

Where

$r_{a,i}$ - Rating assigned to item i by user a,

r_a - Mean rating assigned by user a,

$P_{a,u}$ - Similarity between users a and u, and

m - Number of users in the neighborhood.

The active user's opinion regarding a target item in terms of weighted average of the votes given to that item by other like-minded users is computed. Pearson correlation is used as a similarity measure between users. The algorithms can be summarized in the following steps:

Step 1: All users are weighted with respect to similarity with the active user. Similarity between users is measured as the Pearson correlation between their ratings vectors.
Step 2: Select n active users that have the highest similarity.

Step 3: Compute a prediction, $P_{a,u}$ from a weighted combination. The similarity between two users is computed using the Pearson correlation coefficient:

$$P_{a,u} = \frac{\sum_{i=1}^{rn}(r_{a,i} - \bar{r}_a)\times(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{rn}(r_{a,i} - \bar{r}_a)^3 \times \sum_{i=1}^{rn}(r_{u,i} - \bar{r}_u)^3}} \qquad (4)$$

where

$r_{a,i}$ - Rating assigned to item i by user a and

r_a - Mean of the ratings assigned by user a.

In step 3, predictions are computed as the weighted average of deviations from the neighbor's mean:

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n}(r_{u,i} - \bar{r}_u)\times P_{a,u}}{\sum_{u=1}^{n} P_{a,u}} \qquad (5)$$

Where

$P_{a,i}$ - Prediction for active user a for item i,

$P_{a,u}$ - Similarity between users a and u and

n - Number of users in the neighborhood.

The computation of the predictions, the mean of the predictions of the active users and the actual ratings can be computed using the MAE values, and the results are tabulated with respect to the nearest neighbor set.

5 Singular Value Decomposition (SVD) Algorithm

Singular value decomposition (SVD) [9] plays a vital role in numerical linear algebra and in many statistical techniques. Using two orthonormal matrices, SVD can diagonalize any matrix A, and the results of the SVD can reveal a lot about the consequences of the matrix. There are, of course, various modifications and adjustments to make in order for the algorithm to perform slightly better. These include regularization, using different functions for rating prediction instead of simply taking a dot product of the preference and the feature vector, rounding off in a clever manner, etc. Similarly, SVD is capable of selecting the secular equation to be solved.

5.1 Algorithm Required. Average ratings, the given ratings convert into a matrix of ratings R. compute an approximate matrix R_{app} such that the MAE is minimized

Step 1: Task. Find the best dictionary to represent the data samples as sparse compositions.
Step 2: Initialization. Set the dictionary matrix D. Set J = 1.
Repeat until convergence.
Step 3: Sparse Coding Stage. Use any pursuit algorithm to compute the representation vectors.
Step 4: Update Stage. For each column k = 1, 2... K in D.
Step 5: Compute the overall representation error matrix
Step 6: Restrict E by choosing only the columns corresponding to k.
Step 7: Apply SVD decomposition. Choose the updated dictionary column.
Step 8: Update the coefficient vector multiplied vectors

$$MAE = \left\| R - R_{app} \right\| F$$

Step 9: Compute P as $US^{1/2}$ and F as $S^{1/2}V^{`T}$
Step 10: Minimize the error: $E = \left(R - R_{app}\right)_{ij}^{\;2}$
Step 11: Compute $P_{ik}\left(t+1\right)$ and $F_{jk}\left(t+1\right)$

Taking the derivative with respect to P_{ik} and F_{jk} the updates are

$$P_{ik}\left(t+1\right) = P_{ik}\left(t\right) + L \times \left(R - R_{app}\right)_{ij} \times F_{jk}\left(t\right) - P_{ik}\left(t\right)$$

$$F_{jk}\left(t+1\right) = F_{jk}\left(t\right) + L \times (R - R_{app})_{ij} \times P_{ik} - P_{ik}\left(t\right)$$

The MAE values are computed using the SVD algorithm for nearest neighbor sets, and the results are tabulated.

6 Proposed Content-Boosted Collaborative Filtering (CBCF) Algorithm

In content boosted collaborative filtering [11], Similarity between the active users is computed by using the Pearson correlation coefficient. The following steps describe the proposed algorithm.

Required: Set of items and average ratings.
Step1: A pseudo user-rating vector is created for all users in the database.
Step2: Compute the pseudo-rating matrix by combining the pseudo user-rating vectors of all users.
Step3: Compute the similarity between active user ‘a’ and inactive user ‘u’ using the Pearson correlation coefficient.
Step4: Compute mean-centered ratings of the best n neighbors of the corresponding user as the weighted sum of the active user.
Step5: Combine steps 3 and 4 to evaluate the predictions.

7 Experiments and analysis

The above-mentioned CF algorithms to address the cold-start and data sparsity problems are evaluated using MAE values. The experimental dataset, MovieLens, comes from the movie recommendation system designed by the GroupLens Research Project at the University of Minnesota [10].

7.1 Dataset

The proposed implementation is evaluated using the MovieLens [10] dataset. The dataset consists of 100,000 ratings with ranging from 1 to 5 of 1,682 movies by 943 users. Each user has rated approximately 20 movies. The density of the MovieLens dataset is 0.063.The dataset provides the actual rating data for each user for various movies. User ratings range from 0to5 rating scale. The data have been cleaned up by removing from the dataset users with fewer than 20 ratings, as well as those within complete demographic information. 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.test,u2.test,u3.test,u4.test and u5.test have disjoint test sets; this if for 5 fold cross validation to recur experiment with each training and test set and to get the results for optimum values.

7.2 Performance Evaluation

The prediction quality of a CF approach can be evaluated by comparing recommendations to a test set of known user ratings. These systems are typically evaluated using predictive accuracy metrics, where the predicted ratings are directly compared to actual user ratings. The most commonly used metrics for prediction accuracy are the Mean Absolute Error (MAE). It computes the average of the absolute difference between the predictions and the true ratings. Lower MAE values indicate better predictions.

$$MAE = \frac{\sum_{\{u,i\}} \left| p_{u,i} - r_{u,i} \right|}{n} \tag{6}$$

Where
n - Total number of ratings by all users,
$P_{u,i}$ - predicted rating for item u by user i and
$r_{u,i}$ - Actual rating.

The influence of various nearest neighbor sets on predictive accuracy was tested by gradually increasing the number of neighbors. The item ratings of the users are evaluated according to the opinions of the users chosen ratings.

7.3 Results and Analysis

The results of MAE values for different sizes of neighboring sets are shown in Table 1.The derived MAE values for different test datasets from u1.test to u5.test are related by their recommendation accuracy, which was computed and compared. The MAE was obtained for each component of fivefold cross validation experiment. The total MAE was then computed from the entire set of users and folds in the experiments. The MAE values obtained by using pure CF, the UBCF, the SVD and the CBCF algorithms are listed in Table 1. It is thus evident that the CBCF algorithm reduces the cold start and data sparsity problems in recommender systems.

8 Conclusions

Models of hybrid recommendation systems provide good quality predictions in instances of data sparsity. The modified CBCF system proposed in this paper converts a sparse user rating matrix into a full rating matrix for content data. CBCF algorithm showed the best prediction quality when compared with the pure CF, UBCF and SVD algorithms, and can thus reduce cold start and data sparsity problems. In this paper, it is showed that hybrid collaborative filtering is significantly better than any of the other collaborative filtering approaches. It overcomes some of the disadvantages of both collaborative filtering and content-based filtering by strengthening collaborative filtering with content-based collaborative filtering.

Acknowledgments

Author would like to thank the GroupLens Research Project agency for providing the data used for this study.

References

[1] M.A. Hameed, O. Al Jadaan, and S. Ramachandram, “Collaborative Filtering Based Recommendation System: A survey, “*International Journal of Computational Science and Engineering*, ISSN 0975-3397, Vol.4 No.5 May 2012.

[2]X. Su and T.M. Khoshgftaar, “A Survey of Collaborative Filtering Techniques,”*Advances in Artificial Intelligence*, vol. 2009, article ID 421425, doi:10.1155/2009/421425.

[3] X. Lam, T. Vu, T. Le and A. Duong, “Addressing Cold-Start Problem in Recommendation Systems,” *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*, ACM, Suwon, Korea, (2008) January 31-February 01.

[4]D. Sun, Z. Luo, and F. Zhang, “A Novel Approach for Collaborative Filtering to Alleviate the New Item Cold-Start Problem, “*Communications and Information Technologies* (ISCIT), 2011 *11th International Symposium*, IEEE, Cordoba, Spain, (2011) November 22-24, pp. 402-406.

[5]A. Sunil Kumar, M.S. Prasad Babu and B. Raja Sarath Kumar, “Implementation Hybrid System based on Content-boosted Collaborative Filtering Algorithm, “*Global Journal ofComputational Intelligence Research*. ISSN 2249-0000 Volume 3, Number 1 (2013), pp. 11-20.

[6] Maddali Surendra Prasad Babu and Boddu Raja Sarath Kumar, “An Implementation of the User-based Collaborative Filtering Algorithm, “*International Journal of Computer Science and Information Technologies* (IJCSIT), vol.2 (3), 2011, ISSN 0975-9646, 1283-1286.

[7]L.Zheng, Y. Wang, J.Qi, and D. Liu, “Research and Improvement of PersonalizeRecommendation Algorithm based on Collaborative Filtering, “*International Journal of Computer Science and Network Security*, vol.7, no.7, pp. 134-138, July 2007.

[8] S. Xiaoyuan and M.K. Taghi, “Collaborative Filtering for Multi-class Data Using Belief Nets Algorithms,” *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence* (ICTAI'06) 0-7695-2728-0/2006, 0-7695-2728-0/06, IEEE, 2006.

[9]Serdar Sali, “Movie Rating Prediction Using Singular Value Decomposition,” CMPS 242, Machine Learning Project Report, University of California, Santa Cruz, 2008.

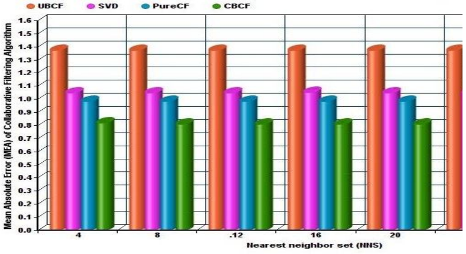
[10]MovieLens data, http://movielens.umn.edu.

[11]Prem Melville, Raymond J. Mooney and 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.

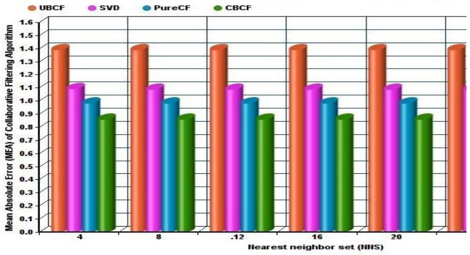
Table 1. MAE values for UBCF, SVD, Pure CF and CBCF

Test Dataset	NNS /MAE	4	8	12	16	20	24	28
U1.test	UBCF	1.370	1.370	1.370	1.370	1.370	1.370	1.370
	SVD	1.049	1.049	1.049	1.050	1.049	1.049	1.049
	Pure CF	0.982	0.982	0.982	0.982	0.982	0.982	0.982
	CBCF	0.820	0.810	0.810	0.810	0.810	0.810	0.810
U2.test	UBCF	1.390	1.390	1.390	1.390	1.390	1.390	1.370
	SVD	1.091	1.090	1.090	1.090	1.090	1.090	1.090
	Pure CF	0.982	0.982	0.982	0.982	0.982	0.982	0.982
	CBCF	0.860	0.860	0.860	0.860	0.860	0.860	0.860
U3.test	UBCF	1.380	1.390	1.390	1.390	1.390	1.380	1.390
	SVD	1.091	1.091	1.091	1.091	1.091	1.091	1.091
	Pure CF	0.982	0.982	0.982	0.982	0.982	0.982	0.982
	CBCF	0.890	0.880	0.880	0.880	0.880	0.880	0.880
U4.test	UBCF	1.390	1.390	1.390	1.390	1.380	1.370	1.370
	SVD	1.161	1.161	1.161	1.161	1.161	1.161	1.161
	Pure CF	0.982	0.982	0.982	0.982	0.982	0.982	0.982
	CBCF	1.161	1.161	1.161	1.161	1.161	1.161	1.161
U5.test	UBCF	1.390	1.390	1.390	1.390	1.390	1.400	1.400
	SVD	1.168	1.168	1.167	1.167	1.167	1.167	1.167
	Pure CF	0.982	0.982	0.982	0.982	0.982	0.982	0.982
	CBCF	0.940	0.940	0.940	0.940	0.940	0.940	0.940

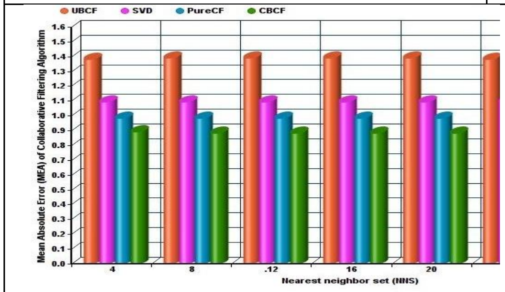
Table 1- Nearest neighbor set (NNS) Vs Mean Absolute Error (MAE) values of User-based Collaborative Filtering (UBCF) Algorithm, Singular Value Decomposition (SVD) Algorithm, Pure Collaborative Filtering and Content-boosted Collaborative Filtering (CBCF) Algorithm with U1.test to U5.test datasets of MovieLens dataset.



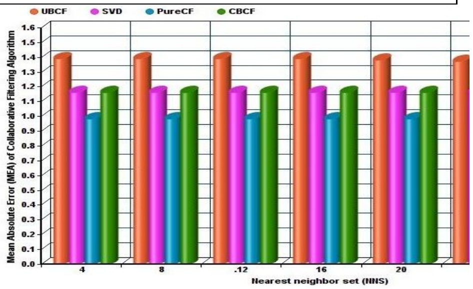
Nearest neighbor set (NNS) vs. Mean Absolute Error (MAE) values of User-based Collaborative Filtering (UBCF) Algorithm, Singular Value Decomposition (SVD) Algorithm, Pure Collaborative Filtering and Content-boosted Collaborative Filtering (CBCF) Algorithm with U1.test datasets of MovieLens dataset.



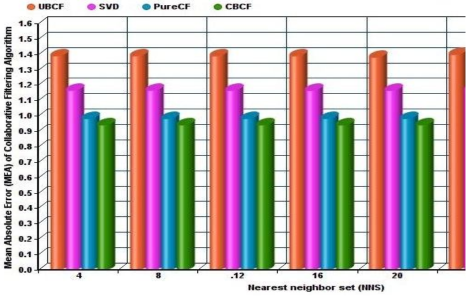
Nearest neighbor set (NNS) vs. Mean Absolute Error (MAE) values of User-based Collaborative Filtering (UBCF) Algorithm, Singular Value Decomposition (SVD) Algorithm, Pure Collaborative Filtering and Content-boosted Collaborative Filtering (CBCF) Algorithm with U2.test datasets of MovieLens dataset.



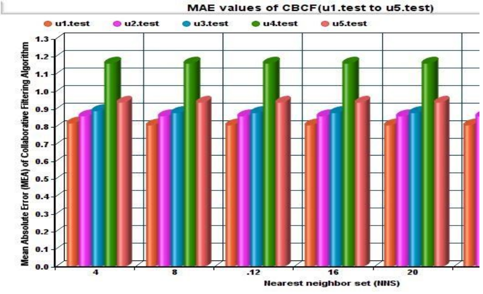
Nearest neighbor set (NNS) vs. Mean Absolute Error (MAE) values of User-based Collaborative Filtering (UBCF) Algorithm, Singular Value Decomposition (SVD) Algorithm, Pure Collaborative Filtering and Content-boosted Collaborative Filtering (CBCF) Algorithm with U3.test datasets of MovieLens dataset.



Nearest neighbor set (NNS) vs. Mean Absolute Error (MAE) values of User-based Collaborative Filtering (UBCF) Algorithm, Singular Value Decomposition (SVD) Algorithm, Pure Collaborative Filtering and Content-boosted Collaborative Filtering (CBCF) Algorithm with U4.test datasets of MovieLens dataset.



Nearest neighbor set (NNS) vs. Mean Absolute Error (MAE) values of User-based Collaborative Filtering (UBCF) Algorithm, Singular Value Decomposition (SVD) Algorithm, Pure Collaborative Filtering and Content-boosted Collaborative Filtering (CBCF) Algorithm with U5.test datasets of MovieLens dataset.



Nearest neighbor set (NNS) vs. Mean Absolute Error (MAE) values of User-based Collaborative Filtering (UBCF) Algorithm, Singular Value Decomposition (SVD) Algorithm, Pure Collaborative Filtering and Content-boosted Collaborative Filtering (CBCF) Algorithm with U1.test dataset to U5.testdataset of MovieLens dataset.