Global Journal of Computational Intelligence Research.
ISSN 2249-0000 Volume 3, Number 1 (2013), pp. 11-20

http://www.ripublication.com/gjcir.htm

Implementation Hybrid System based on Content Boosted Collaborative Filtering Algorithm

A. Sunil Kumar*, Prof. M. S. Prasad Babu
and B. Raja Sarath Kumar*****

M. Tech (CN), Andhra University, Visakhapatnam
sk75kumar@gmail.com
Dept. of CS & SE, Andhra University, Visakhapatnam
msprasadbabu@yahoo.co.in
Professor, Dept. of CSE,
Lenora College of Engineering, Rampachodavaram
iamsarathphd@gmail.com

Abstract:

Recently, the Web, globally and within intranets, has been expanding both in the size of the information space and in the number of the users of that space. As a result, the task of locating relevant information and navigating that space in order to make choices of good items has been providing more and more difficult Most recommender systems use Collaborative Filtering (CF) or Content-based (CB) methods to predict new items of interest for a user. While both methods have their own advantages, but individually they fail to provide good recommendations in many situations., a hybrid recommender system can overcome these short-comings, by incorporating components from both content and collaboration methods and combine them These approaches use a content-based predictor to enhance existing user data, and then provides personalized suggestions through collaborative filtering So Content Boosted Collaborative Filtering performs better than a pure collaborative filter and pure content-based predictor, and also improve the performance of this hybrid system.

Keywords: Recommender Systems, Collaborative Filtering (CF), Content based (CB), Movie lens, Naïve Bayes, Pearson Correlation Coefficient, neighborhood, MAE

1. Introduction:

There have been efforts to automate filtering of relevant information and presenting organized information to the user. Such automated methods, commonly referred to as intelligent information retrieval are used to locate and retrieve information with respect to a user‘s individual preferences [1]. One key area to achieve this goal is targeted around *recommender systems* (RS) An RS is an agent-based system. Recommender systems solve information overload by using personalized suggestions based on a history of a user's likes and dislikes. As all on-lines provide recommending services in an ever-increasing number of e-commerce sites such as Amazon [2] for books, Movie Lens [3], for movies, There are two different approaches to build recommender systems are Content-based (CB) Collaborative Filtering (CF) recommending.

1.1 Content-Based System

This approach recommends items that are similar in content to items the user has liked in the past, or matched to attributes of the user.The main assumption under content-based approaches is that an item or document can be identified by a set of features extracted directly from their content. content-based recommendation techniques are special cases of information-based techniques; in the way that they attempt to make predictions based on the analysis of the items associated with them In [4] look at all items a user has rated in the past and determines how similar they are to the current item. For those items that are similar enough, the old ratings are used to calculate a predicted rating for the unrated item. Provide recommendation by comparing representations of content contained in an item to the content that interests the user

1.2 Collaborative System

CF is the method which automatically predicts the interest of an user by collecting rating information from other similar users or items, without any descriptive data in a given domain and make differences and similarities among several users (or) items in determining how to recommend. The underlying assumption of CF is that the active user (the user that the prediction refers to) will prefer those items which the similar users prefer. Once all the user profiles have been collected, the active user‘s similarities with the remaining of the users are calculated. Similarity is usually computed to user/item by comparing rating-vectors with distance metrics, e.g. Pearson correlation while in users similarity the group (neighborhood) of the users that are most similar to the active user is selected and their ratings are combined to produce predictions. Predictions of ratings may typically lead to the presentation of a top-n-list of the most relevant items [5]

CF has some great advantages over CB. Firstly, CF can perform in domains where the content is difficult for a computer to analyze ideas; opinions etc. next CF system has the ability to recommend items that are relevant to the user, but not the content from user’s profile. Because of these issues, CF systems have been used successfully to build recommender systems

in various applications [6].However CF suffer from two problems

The process of finding better neighbors among the users is a challenge if users are rated to very less movies. This leads to Sparsity states that 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, and also a First-rater Problem where movie cannot be recommended unless a user has rated it before.

To overcome the CF drawbacks, a new hybrid approach Content-Boosted Collaborative Filtering (CBCF) is used in movie recommendations and show, this approach performs better than both pure CF and CB systems

2. SYSTEM DESCRIPTION

The URLs provided in the Movie lens dataset downloaded movie content from IMDB.The Movie lens dataset provides the user-ratings matrix; of users versus items, each row of this matrix is a user ratings vector. Where most items have not been rated by most users. The user-ratings matrix is very sparse, it then fills in the missing values of the rating matrix with the predictions of the content predictor to form a pseudo rating matrix, in which user ratings are kept untouched and missing ratings are replaced by the content predictor. It then makes predictions over the resulting pseudo ratings matrix using a weighted Pearson correlation-based CF algorithm, and then an active user's predictions are made for a new item using CF on full pseudo ratings matrix.The overview of system is shown in Fig1

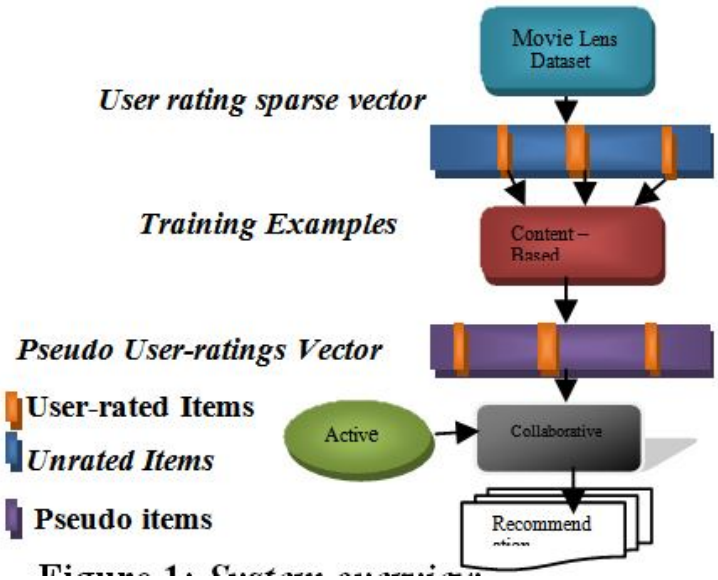

Figure 1: *System overview*

2.1 Movie Lens Dataset

To demonstrate the working of hybrid approach in the domain of movie recommendation, a user-movie rating is provided by the Movie lens dataset. The contents of each movie are downloaded from IMD [7] The representation of the content information of every movie is as a set of slots (features). Each slot is represented simply as a bag of words. The slots that use for the Movie lens dataset are: movie title and type.

2.2 Pure Content based Predictor

The content-boosted CF algorithm uses naive Bayes as the content classifier, it then fills in the missing values of the rating matrix with the predictions of the content predictor to form a pseudo rating matrix A naive Bayes classifier requires a small amount of training data to estimate the parameter necessary for classification[8].

To provide content-based predictions the prediction task is treated as a text-categorization problem and the movie content information as text documents, and user ratings 1-5 as one of five class labels. A bag-of-words naive Bayesian text classifier [9] is implemented to learn a user profile from a set of rated movies i.e. labeled documents.

A multinomial text model [10], document is used as sequence of words drawn from the same vocabularyV. The naive Bayes assumption states that the probability of each word event is dependent on the document class but independent of the word's context and position. For each class c_j, and word (token), $\omega_k \in$V, the probabilities, P (c_j) and P ($\omega_k|c_j$) must be observed from the training data. Then the posterior probability of each class given a document D is computed using Bayes rule

$$P\ (c_j|D)=\frac{P\ (c_j)}{P(D)}\ \Pi_{i=1}^{|D|}P(a_i|c_j) \quad (1)$$

|D| is the number of words in the document and where a_i is the i^{th} word in the document. In the document P(D) can be neglected since it is constant, documents d_m represent movies vector one for each slot, P($\omega_{k|}|c_j$, s_m) is the probability of given the category and the slot of each word must be estimated for a movie, F computed using the posterior category probabilities,

$$P(c_j|F)=\frac{P\ (c_j)}{P\ (F)}\Pi_{m=1}^{S}\Pi_{i=1}^{|d_m|}P\ (a_{mi}|c_j,s_m) \quad (2)$$

Where S is the number of slots. (s_m denotes the m^{th}slot) and a_{mi} is the i^{th} word in the m^{th} slot. The class with the highest poster.ior probability determines predicted rating. Laplace smoothing [11] is used to avoid zero probability estimates.

Algorithm 1: Content Based Predicator

Train Naive Bayes (Movies, C)

Each movie in Movies is a vector of bag-of-words and a category

corresponding to a 1-5 rating. Each bag of bag-of-words corresponds to a slot like title. This function estimates the probability terms P ($a_{mi}|c_j$, s_m), describes the probability that a randomly drawn word from a slot s_m in a movie in class c_j will be the word a_{mi} is value for each class label j which are subsets from Movies documents

1 To calculate class priors, P (c_j)

$ratings_j$ is value for each class label j

Which are subsets from Movies document

P(c_j) $\leftarrow \frac{|\text{ratings}_j| + \frac{1}{|Movies|}}{|Movies| + \frac{|C|}{|Movies|}}$ C is the set of all possible classes

2 2 Calculate conditional probabilities,

P ($a_{mi}|c_j$, s_m) for each slot s_m,

$\text{Vocabulary}_m \leftarrow$ set of all distinct words

occur in slot s_m all movies

For each possible class c_j n←total number of distinct word positions in slot s_m in the class c_j

For each word, a_{mi} in Vocabulary_m

$n_k \leftarrow$ Number of times same word a_{mi} occurs in slot s_m in class c_j

P ($a_{mi}|c_j, s_m$)$\leftarrow \frac{n_k + \frac{1}{|Movies|}}{n + \frac{|\text{Vocabulary}_m|}{|Movies|}}$

2.3 Pure Collaborative Filtering

A pure collaborative filtering component is implemented using a neighborhood-based algorithm [12]. The algorithm produces recommendations list for active user according to the view of other users. CF recommendation system use statistical techniques to search the nearest neighbors of the active user and then basing on item rating rated by the nearest neighbors to predict the item rating rated by active user, then produces recommendation list

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 follows as 1 Weight all users with respect to similarity with the active user.

Similarity between two users is measured as the Pearson correlation between their rating vectors.

$$P_{a,u} = \frac{\sum_{i=1}^{m}(r_{a,i} - \bar{r_a}) \cdot (r_{u,i} - \bar{r_u})}{\sqrt{\sum_{i=1}^{m}(r_{a,i} - \bar{r_a})^2 \cdot \sum_{i=1}^{m}(r_{u,i} - \bar{r_u})^2}} \quad (3)$$

$r_{a,i}$ is the rating to item i by user a; $\bar{r_a}$ is mean rating by user a.

Select n users that have the highest similarity with the active user.
These users form the neighborhood and n users are selected, that have the highest profile similarity with active user *a*. These users constitute the neighborhood of active user a

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

The weighted predictions are computed as the weighted average of deviations from the neighbor's mean:

$$P_{a,i}=\bar{r}_a + \frac{\sum_{u=1}^{n}(r_{u,i}-\bar{r}_u)*p_{a,u}}{\sum_{u=1}^{n}p_{a,u}} \quad (4)$$

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. n is the number of users in the neighborhood. n is the number of users in the neighborhood. Here neighborhood size is of 25, [15].

Collaborative filtering would not perform well, if a high correlation with his neighbors was based on very few co-rated items. To eliminate those bad predictors the correlation is multiplied with a *significance weighting* factor $sg_{a,u}$; $n_{a,u}$ is the number of items users a and u have co-rated.

$$sg_{a,u} = \begin{cases} \frac{n_{a,u}}{50} : if\ n_{a,u} < 50 \\ 1 : otherwise \end{cases} \quad (5)$$

3 Content-Boosted Collaborative Filtering

CBCF developed to overcome the problems with CF and CB. The Content-Based predictions used to "convert a sparse user ratings matrix into a full user ratings matrix." creating a pseudo user ratings vector for every user u in the dataset. 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.

$$\upsilon_{u,\ i}=\begin{cases} q_{u,i} : if\ user\ u\ rated\ item\ i \\ p_{u,i} : otherwise \end{cases}$$

In the above equation $q_{u,\ i}$ denotes the actual rating provided by user u for item i, while $p_{u,\ i}$ is the rating predicted by the pure content-based system gives the dense pseudo ratings matrix V.Then apply the collaborative filtering on this dense matrix to compute recommendations, similarity between the active user and another user is computed using the Pearson correlation coefficient described in Equation3

3.1. Harmonic Mean Weighting

To have fair correlations between active user and other users. Pseudo user-ratings vector depends on the number of movies user has rated. If the user rated many items, the content-based predictions are good and hence his pseudo user-ratings vector is fairly accurate. If the user rated only a few items, the pseudo user-ratings vector will not be as accurate. Then inaccuracy shows

misleading. So to incorporate accuracy in the correlations, weights them using the Harmonic Mean weighting factor (HM weighting).

$$hm_{i,j} = \frac{2 m_i m_j}{m_i + m_j} \; ; \; m_i = \begin{cases} \frac{n_i}{50} : if \; n_i < 50 \\ 1 : otherwise \end{cases}$$

n_i No of items user i rated

The harmonic mean tends to bias the weight towards the lower of the two values m_i and m_j.To the HM weight, add the significance weighting described in equation 5, and thus obtain the hybrid correlation weight $hw_{a,u}$

$$. \; hw_{a,u} = hm_{a,u} + . \; sg_{a,u} \tag{6}$$

3.2 Self Weighting

Prediction for the active user is computed as a weighted sum of the mean-centered votes of the best-n Neighbors of that user. Pseudo active user is added to the neighborhood, to increase the confidence of pure-content predictions for the active user. By incorporating a Self Weighting factor in the final prediction:

$$sw_a = \begin{cases} \frac{n_a}{50} * max : if \; n_a < 50 \\ max : otherwise \end{cases} \tag{7}$$

n_a is the number of items rated by active user value for max is 2.

3.3 Producing Predictions to active user

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

$$p_{a,i} = \overline{v_a} + \frac{sw_a (c_{a,i} - \overline{v_a}) + \sum_{u=1, u \neq a}^{n} hw_{a,u} p_{a,u} (v_{u,i} - \overline{v_u})}{sw_a + \sum_{u=1, u \neq a}^{n} hw_{a,u} p_{a,u}} \tag{8}$$

$c_{a,i}$ 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 and $\overline{v_u}$ is the mean over all items for that user. sw_a, $hw_{a,u}$ and $p_{a,u}$ are in Equations 7, 6 and 3 respectively; n is the size of neighborhood. The denominator is normalization factor ensures all weights sum to one.

4 Results

```
Enter user id
3
results
neighbor for the user 3 is 284
neighbor for the user 3 is 61
neighbor for the user 3 is 920
neighbor for the user 3 is 724
neighbor for the user 3 is 866
neighbor for the user 3 is 809
neighbor for the user 3 is 772
neighbor for the user 3 is 149
neighbor for the user 3 is 923
similarity between user 3 neighbor 284    0.36143059734454563
similarity between user 3 neighbor 61    0.26895643797839974
similarity between user 3 neighbor 920    0.2351851297486829
similarity between user 3 neighbor 724    0.2327523581973842
similarity between user 3 neighbor 866    0.23266626819870198
similarity between user 3 neighbor 809    0.23120669331403554
similarity between user 3 neighbor 772    0.23060923460318192
similarity between user 3 neighbor 149    0.21697636036482537
similarity between user 3 neighbor 923    0.21388511476196032
neighbor rating on item 1 is  2.0
neighbor rating on item 1 is  5.0
neighbor rating on item 1 is  4.0
neighbor rating on item 1 is  3.0
neighbor rating on item 1 is  2.0
neighbor rating on item 1 is  4.0
neighbor rating on item 1 is  3.0
neighbor rating on item 1 is  4.0
Prediction for the movie id 1 is  3.871745465825128
```

To obtain results most widely used metrics in CF are available.For accuracy of a prediction algorithm statistical accuracy is used. Statistical accuracy metrics evaluate the accuracy of a predictor by comparing predicted values with user-provided values. To measure statistical accuracy we use the mean absolute error (MAE) metric which computes the average of the absolute difference between the predictions and true ratings

$$\text{MAE}=\frac{\sum_{(i,j)} |p_{i,j}-r_{i,j}|}{n}$$

Where n is the total number of ratings over all users, $p_{i,j}$ is the predicted rating for user i on item j, and $r_{i,j}$ is the actual rating. The lower the MAE, better the prediction.

Table 1: MAE

Neighbor size	5	10 10	15	20	25	30
MA E	0.92	0.89	0.867	0.835	0.821 0.821 0.821 0.821	0.793

Table 2: Summary of Results

Algorithm	MAE
Content-based	0.913
Collaborative filtering	0.837
Content-boosted CF	0.791

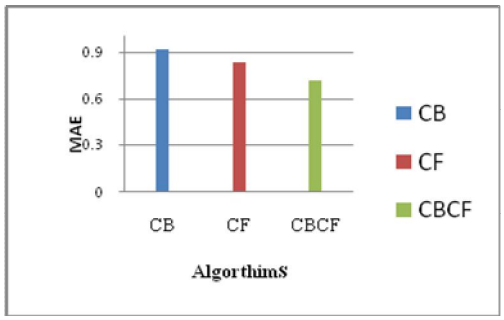

Figure 2 Comparison Of Algorithms

5 CONCLUSIONS

Appling content information into collaborative filtering can significantly improve predictions of a recommender system. In this paper we have shown how Content-boosted Collaborative Filtering performs significantly better than a pure content-based predictor and collaborative filtering, It overcomes the disadvantages of both collaborative filtering and content-based methods, by support of CF with content and vice versa. Further, due to the nature of the approach, any improvements in collaborative filtering or content-based recommending can be easily exploited to build a more powerful system.

6 References

[1] M. Balabanovic, ―Learning to Surf: Multiagent systems for adaptive web page recommendation, ‖ Ph.D Thesis, Stanford University, Stanford, CA, 1998

[2] Amazon.com, www.amazon.com, Last accessed on August 2009.

[3] MovieLens, www.movielens.umn.edu, last accessed on August 2009.
[4] Nicola Orio, ―Music Retrieval: A tutorial and Review, ‖ *Foundations and Trends in Information Retrieval*, Vol. 1, Issue 1, pp 1-96, 2006.
[5] Mark Van Setten, ―Supporting people in finding information, Hybrid recommender systems and goal-based structuring, ‖ *Telematica Instituut Fundamental Research Series*, vol. 016. Enschede, the Netherlands: Telematica Instituut, 2005.
[6] George Lekakos, Petros Caravelas, A hybrid approach for movie recommendation, ‖ *Springer Science + Business Media*, LLC, December 2006.
[7] The Internet Movie Database (IMDb), http://www.imdb.com/, Last accessed August 2009.
[8] D. Goldberg, D. Nichols, B. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. Communications of the Association of Computing Machinery, 35(12):61{70, 1992.
[9] T. Mitchell. Machine Learning. McGraw-Hill, New York, NY, 1997.
[10] A. K. McCallum and K. Nigam. A comparison of event models for naive Bayes text classication. In Papers from the AAAI 1998 Workshop on Text Categorization, pages 41{48, Madison, WI, 1998.
[11] R. Kohavi, B. Becker, and D Sommer_eld. Improving simple Bayes. In Proceedings of the European Conference on Machine Learning, 1997
[12] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative _ltering. In SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 230{237, 1999.