Volume:04 Issue:02 Pages: 1568-1573 *(2012) ISSN : 0975-0290.*

A Study of Collaborative Filtering Approach for Temporal Dynamic Web Data

Meghna Khatri
Department of Computer Science and Engineering, Maharishi Dayanand University, Rohtak
Email: meghna823@gmail.com

----------ABSTRACT----------

Collaborative filtering is widely used and popular tool these days. In collaborative filtering, user preference data, collected over a long period of time, is exploited to predict interest on the unseen items the basis of users with similarity interests. The similarity amongst the items is determined by the similarity function as weighted average of the ratings given by the users. In this paper, an improved similarity function for collaborative filtering is proposed that incorporates the time when the item was rated. This allows the collaborative filtering to capture the data more accurately and efficiently.

Keywords – **collaborative filtering, temporal dynamics, web data**

Date of Submission : August 01, 2012 Date of Acceptance : August 31, 2012

1. INTRODUCTION

The collaborative filtering has been proven useful and has been widely used in many areas, for example recommender systems for movies, music, web pages, news, Usenet articles, TV programme and e-commerce. It helps people to receive personalised recommendations and lightens the burden of users in the explosive information stream as well as enhances the sale volume of e-commerce web sites.

In collaborative filtering user preference data is exploited to predict interest on the unseen items. These are mechanisms that attempt to predict items in which a user might be interested, given some information about the users' with similar interests. That is to say, given the information of other users' rating of the items, a new prediction is provided to the user that has never been evaluated before and would likely to be interested in.

However, traditional collaborative filtering does not take the changing behaviour of each user's interests into account and uses old data to predict new data on the basis of users' with similar interests. This is similar to the traditional word-of-mouth behaviour. The similarity amongst the items is determined by the similarity function as weighted average of the ratings given by similar users, where the weight is proportional to the user similarity. Therefore, the accuracy of similarity is the key to success of collaborative filtering.

The collaborative filtering uses the given large amount of user feedbacks such as ratings, clicks purchases, etc., collaborative filtering algorithm works by discovering the similarity between users and items and predict unobserved ratings based on the observed ratings associated with similar users and items. In real world applications, user feedback data are accumulated over time as users interact with the system, the data instances can be naturally ordered by the time they are collected.

Most existing approaches often ignore the dimension of time and assume that the users' and the items' characteristics are static. While such assumption is acceptable for relatively short time periods such as days or weeks, it becomes rather unreasonable for longer time periods during which important factors affecting recommendation decisions such as a users' interests or a movie's popularity can vary significantly. There are many causes of such causes of such changes. Firstly, a user's interests or tastes often change over time. Secondly, external events such as holidays could lead to abrupt increase in the popularity of certain items such as comedies. Thirdly, as time goes by, the recommender system itself may went through changes such as reorganizing its catalogue or introducing new search or linking features, which may improve the accessibility of some items. Finally, a user's behaviour often exhibit temporal locality. For example, if a person enjoyed a particular movie, he will often try to find related movies by the same directors/actors or of the same genre.

But space was left for improvement in order to provide better performances if the systems can know its users in every detail, it can provide more precise results but most algorithms do not take the temporal factor into account especially when considering the similarities between users and items. The temporal information included at the time of calculating similarity may affect the collaborative filtering performance very much as the more recent evaluations reflect the users' current interests better.

In order to inculcate all this temporal information, we introduce a time function that gives weightage to the ratings according or in an order they were rated. The expected output is that the new algorithm will perform better and will provide better results when evaluated against the various evaluation metrices.

2. BACKGROUND

2.1. Collaborative Filtering Approach

The Collaborative filtering refers to a process for predicting item preferences based on the preferences of other users with the similarity behavior. As one of the most successful technologies for many popular applications, it has been widely developed and improved over the past decade.

Volume:04 Issue:02 Pages: 1568-1573 *(2012) ISSN : 0975-0290.*

The collaborative filtering technique applied to recommender systems matches people with similar interests and then makes recommendations based on this basis. Recommendations are commonly extracted from the statistical analysis of patterns and analogies of data extracted explicitly from evaluations of items (ratings) given by different users or implicitly by monitoring the behavior of the different users in the system.

Collaborative filtering is very different from content-based filtering, the other most commonly used approach in recommender systems. Rather than recommending items because they are similar to items the user has liked in the past, items are recommended based on other users' preferences. Rather than computing the similarity of items, the similarity among users is computed. In collaborative filtering a user's profile consists simply of the data the user has specified. This data is compared to those of other users to find overlaps in interests among users. These are then used to recommend new items. Typically, each user has a set of nearest neighbours defined by using the correlation between past evaluations. Predicted Scores for un-evaluated items of a target user are predicted by recommender system using a combination of the actual rating scores from the nearest neighbours of the target user.

The most important three essentials are needed to support collaborative filtering: many people must participate to increase the likelihood that any one person will find other users with similar preferences, there must be an easy way to represent a user's interests in the system, and the algorithms must be able to match people with similar interests. The first element is not easy to supply because it needs to change people's using habits and result in the main shortcoming of collaborative filtering systems:

- The early-rater problem: When a new item appears in the database, there is no way it can be recommended to a user until more information is obtained through another user either rating it or specifying which other items it is similar to.

- The sparsity problem: The goal of collaborative filtering systems is to help people focus on reading documents of interest. As with the previous short-coming, if the number of users is small relative to the volume of information in the system, there is a danger of the coverage of ratings becoming too sparse, thinning the collection of recommendable items. Also, sparsity problem poses a real computational challenge as it becomes harder to find neighbors and harder to recommend items since too few people have given ratings.

- The grey sheep problem: This problem is about for some users whose tastes vary from the norm, there will not be any other users who share his or her particular likes and dislikes. This means that, although they have other users for which a high correlation coefficient is calculated, recommendations based on them largely turn out to be false positives.

The goal of a collaborative filtering algorithm is to suggest new items or to predict the utility of a certain item for a particular user based on the user's previous likings and the opinions of other like-minded users. In a typical collaborative filtering scenario, there is a list of m users $U = \{u_1; u_2; : : : ; u_m\}$ and a list of n items $I = \{i_1; i_2; : : : ; in\}$. Each user u_i has a list of items I_{ui} , which the user has expressed his/her opinions about. Opinions can be explicitly given by the user as a rating score, generally within a certain numerical scale, or can be implicitly derived from purchase records, by analyzing timing logs, by mining web hyperlinks and so on [1, 2].

Figure 1 shows the schematic diagram of the collaborative filtering process. CF algorithms represent the entire m ×n user-item data as a ratings matrix, A. Each entry $a_{i;j}$ in A represents the preference score (ratings) of the ith user on the jth item. Each individual rating is within a numerical scale and it can as well be 0 indicating that the user has not yet rated that item. Researchers have devised a number of collaborative filtering algorithms that can be divided into two main categories: Memory-based (user-based) and Model-based (item-based) algorithms [3].

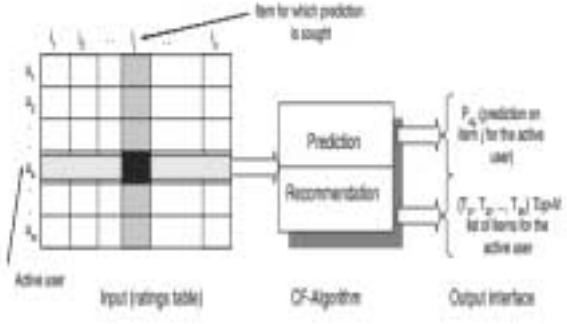

Figure 1: The Collaborative Filtering Process.

In this section we provide a detailed analysis of CF-based recommender system algorithms.

Memory-based Collaborative Filtering Algorithms.

Memory-based algorithms utilize the entire user-item database to generate a prediction. These systems employ statistical techniques to find a set of users, known as neighbours that have a history of agreeing with the target user (i.e., they either rate different items similarly or they tend to buy similar set of items). Once a neighbourhood of users is formed, these systems use different algorithms to combine the preferences of neighbours to produce a prediction or top-N recommendation for the active user. The techniques, also known as nearest-neighbour or user-based collaborative filtering, are more popular and widely used in practice.

Model-based Collaborative Filtering Algorithms.

Model-based collaborative filtering algorithms provide item recommendation by first developing a model of user ratings. Algorithms in this category take a probabilistic approach and envision the collaborative filtering process as

Volume:04 Issue:02 Pages: 1568-1573 *(2012) ISSN : 0975-0290.*

computing the expected value of a user prediction, given his/her ratings on other items. The model building process is performed by different machine learning algorithms such as Bayesian network, clustering, and rule-based approaches. The Bayesian network model [3] formulates a probabilistic model for collaborative filtering problem. Clustering model treats collaborative filtering as a classification problem [4, 3, and 5] and works by clustering similar users in same class and estimating the probability that a particular user is in a particular class C and from there computes the conditional probability of ratings. The rule-based approach applies association rule discovery algorithms to find association between co-purchased items and then generates item recommendation based on the strength of the association between items.[6]

2.2 Study of time factor

The collaborative filtering uses the given large amount of user feedbacks such as ratings, clicks purchases, etc., collaborative filtering algorithm works by discovering the similarity between users and items and predict unobserved ratings based on the observed ratings associated with similar users and items. In real world applications, user feedback data are accumulated over time as users interact with the system, the data instances can be naturally ordered by the time they are collected.

Most existing approaches often ignore the dimension of time and assume that the users' and the items' characteristics are static. While such assumption is acceptable for relatively short time periods such as days or weeks, it becomes rather unreasonable for longer time periods during which important factors affecting recommendation decisions such as a users' interests or a movie's popularity can vary significantly. There are many causes of such causes of such changes. Firstly, a user's interests or tastes often change over time. Secondly, external events such as holidays could lead to abrupt increase in the popularity of certain items such as comedies. Thirdly, as time goes by, the recommender system itself may went through changes such as reorganizing its catalogue or introducing new search or linking features, which may improve the accessibility of some items. Finally, a user's behaviour often exhibit temporal locality. For example, if a person enjoyed a particular movie, he will often try to find related movies by the same directors/actors or of the same genre.

In order to inculcate all this temporal information, introduce a time function that gives weightage to the ratings according or in an order they were rated.

3. Literature Review

3.1 Study of collaborative filtering approach

Collaboration [n.]: The act of working together; cooperating. Selection of items is based on overlap of interests. The approach is similar to giving out recommendations to a friend. For example, I like romantic movies, and I know that my friend X likes some of the same romantic movies. Thus, if I come across a new romantic movie which I like, I recommend it to him/her, and chances of him/her liking it are quite high. A somewhat better analogy: a group of friends working together to decide what gift to buy for another friend's birthday: a fairly complex process if you try to formalize it.

The concept of collaborative filtering descends from the work in the area of information filtering.

The developers of one of the first recommender systems, Tapestry [16] (other earlier recommendation systems include rule-based recommenders and user-customization), coined the phrase "collaborative filtering (CF),"who first to publish in account of using collaborative filtering technique in the filtering of information. They built a system for filtering email called Tapestry which allowed users to annotate message. Annotations became accessible as virtual fields of the message, and uses could construct filtering queries which accessed those fields. Users could then create queries such as "show me all office memos that Bill thought were important". The collaborative filtering provided by Tapestry was not automated and required users to construct complex queries in a special query language designed for the task. The term collaborative filtering has been widely adopted in the field of recommender systems regardless of the facts that recommenders may not explicitly collaborate with recipients and recommendations may suggest particularly interesting items, in addition to indicating those that should be filtered out [17].

The fundamental assumption of CF is that if users X and Y rate n items similarly, or have similar behaviours (e.g., buying, watching, listening), and hence will rate or act on other items similarly [18]. The collaborative filtering technique applied to recommender systems matches people with similar interests and then makes recommendations based on this basis. Recommendations are commonly extracted from the statistical analysis of patterns and analogies of data extracted explicitly from evaluations of items (ratings) given by different users or implicitly by monitoring the behaviour of the different users in the system.

In collaborative filtering a user's profile consists simply of the data the user has specified. This data is compared to those of other users to find overlaps in interests among users. These are then used to recommend new items. Typically, each user has a set of 'nearest neighbours' defined by using the correlation between past evaluations. Predicted scores for un-evaluated items of a target user are predicted by recommender system using a combination of the actual rating scores from the nearest neighbours of the target user [27].

The problem of lack of transparency in the collaborative filtering systems was introduced in [2]. Collaborative systems today are black boxes, computerized oracles which give advice but cannot be questioned. A user is given no indicators to consult in order to decide when to trust a recommendation and when to doubt one. These problems have prevented acceptance of collaborative systems in all

Volume:04 Issue:02 Pages: 1568-1573 *(2012) ISSN : 0975-0290.*

but low-risk content domains since they are untrustworthy for high-risk content domains.

Early generation collaborative filtering systems, such as GroupLens [19], use the user rating data to calculate the similarity or weight between users or items and make predictions or recommendations according to those calculated similarity values. The so-called memory-based collaborative filtering methods are notably deployed into commercial systems because they are easy-to-implement and highly effective [20, 21]. Customization of CF systems for each user decreases the search effort for users. It also promises a greater customer loyalty, higher sales, more advertising revenues, and the benefit of targeted promotions [22].

However, there are several limitations for the memory based collaborative filtering techniques, such as the fact that the similarity values are based on common items and therefore are unreliable when data are sparse and the common items are therefore few.

To achieve better prediction performance and overcome shortcomings of memory-based collaborative filtering algorithms, model-based collaborative filtering approaches have been investigated. Model based collaborative filtering techniques use the pure rating data to estimate or learn a model to make predictions [2]. The model can be a data mining or machine learning algorithm. Well-known model-based collaborative filtering techniques include Bayesian belief nets (BNs) collaborative filtering models [3–5], clustering collaborative filtering models [23, 24], and latent semantic collaborative filtering models [2]. An MDP (Markov decision process)-based collaborative filtering system [6] produces a much higher profit than a system that has not deployed the recommender.

Hybrid collaborative filtering techniques, such as the content-boosted collaborative filtering algorithm [25] and Personality Diagnosis (PD) [26], combine collaborative filtering and content-based techniques, hoping to avoid the limitations of either approach and thereby improve recommendation performance.

Pushing technology to be more and more accurate requires deepening their foundations, while reducing reliance on arbitrary decisions. An interesting outcome is forming surprising links among seemingly different techniques. For example, at their limit, user-user and item-item neighbourhood models may converge to a single model. The quest for more accurate models does not stop at pushing the foundations of the models to their limits. At least as important is the identification of which kinds of signal, or features, are extractable from the data. [28] Conventional, techniques address to sparse data of user-item preferences (or ratings). Accuracy significantly improves by also addressing less obvious sources of information. They can be mainly categorized into two classes:

- Statistical accuracy metrics evaluate the accuracy of a system by comparing the numerical recommendation scores against the actual user ratings for the user-item pairs in the test dataset. Mean Absolute Error (MAE) between ratings and predictions is a widely used metric. MAE is a measure of the deviation of recommendations from their true user-specified values. For each ratings-prediction pair < pi; qi > this metric treats the absolute error between them equally. The MAE is computed by first summing these absolute errors of the N corresponding ratings-prediction pairs and then computing the average. Formally,

$$MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N}$$

  The lower the MAE, the more accurately the recommendation engine predicts user ratings. Root Mean Squared Error (RMSE), and Correlation are also used as statistical accuracy metric.

- Decision support accuracy metrics evaluate how effective a prediction engine is at helping a user select high quality items from the set of all items. These metrics assume the prediction process as a binary operation- either items are predicted (good) or not (bad). With this observation, whether an item has a prediction score of 1:5 or 2:5 on a five-point scale is irrelevant if the user only chooses to consider predictions of 4 or higher. The most commonly used decision support accuracy metrics are reversal rate, weighted errors and ROC sensitivity .

3.2 Study of role of similarity function in CF

Work by Lathia, N. et al [8] gives an outline of a method of how to depict user-similarity over time. In order to incorporate time factor, the user rating is sorted according to when they were input and then simulate a system that iteratively updates (every μ days). Beginning at time (t=ε) all ratings before ε are used to train the algorithm and test on all ratings input before the next update, at time (ε+μ). This process is repeated for each time t, incrementing by μ at each step. At each step, what was previously tested on becomes incorporated into the training set and simulated on the system.

Time-changing baseline predictors was given by Koren, Y [7] in which it is proposed to include the temporal variability within the baseline predictors through two major temporal effects. First is addressing the fact that an item's popularity is changing over time. For example, movies can go in and out of popularity as triggered by external events such as the appearance of an actor in a new movie. This is manifested in our models by the fact that item bias bi will not be a constant but a function that changes over time. The second major temporal effect is related to user biases - users change their baseline ratings over time. For example, a user who tended to rate an average movie "4 stars", may now rate such a movie "3 stars", for various reasons explained earlier.

Volume:04 Issue:02 Pages: 1568-1573 *(2012) ISSN : 0975-0290.*

Hence, in our models we would like to take the parameter b_u as a function of time. This induces the following template for a time sensitive baseline predictor:

$$b_u i(t) = \mu + b_u(t) + b_i(t)$$

The function $b_{ui}(t)$ represents the baseline estimate for u's rating of i at day t. Here, $b_u(t)$ and $b_i(t)$ are real valued functions that change over time.

More work by Koren, Y., Bell, R., [13] which gives a more detailed approach of capturing temporal dynamics with the baseline predictors and also states more prediction rules.

In another work by Lathia, N., [9] a new approach is proposed. They say that by minimizing the mean error produced when predicting hidden user ratings and also if we adopt an approach of adaptive neighbourhoods [20] then root mean square error is considered to be a criterion for including temporal factor.

Another approach of implicit feedback is given by Lee, T.Q., Park, T., [6] which proposes to give the pseudo ratings matrix an entry '1' as a rating value when a user u purchases. A Time-based Pseudo Rating Matrix is generated where two kinds of temporal information are incorporated - the time when the item was launched and the time when the user purchased an item - into the simple pseudo rating matrix. Two observations are taken:

• More recent purchases better reflect a user's current preference.

• Recently launched items appeal more to users.

Based on these observations, they define a rating function *w* that computes rating values (rather than simply assigning 1) as follows:

w (pi, lj) = The rating value when an item with launch time lj was purchased at time pi.

In the work [35], we show that temporal diversity is an important facet of recommender systems, by showing how CF data changes over time and performing a user survey. We then evaluate three CF algorithms from the point of view of the diversity in the sequence of recommendation lists they produce over time. We examine how a number of characteristics of user rating patterns (including profile size and time between ratings) affect diversity. We then propose and evaluate set methods that maximise temporal recommendation diversity without extensively penalising accuracy.

[36], Time-aware recommender systems (TARS) are systems that take into account a time factor - the age of the user data. There are three approaches for using a time factor: (1) the user data may be given different weights by their age, (2) it may be treated as a step in a biological process and (3) it may be compared in different time frames to find a significant pattern. This research deals with the latter approach. When dividing the data into several time frames, matching users becomes more difficult – similarity between users that was once identified in the total time frame may disappear when trying to match between them in smaller time frames.

The user matching problem is largely affected by the sparsity problem, which is well known in the recommender system literature. Sparsity occurs where the actual interactions between users and data items is much smaller in comparison to the entire collection of possible interactions. The sparsity grows as the data is split into several time frames for comparison. As sparsity grows, matching similar users in different time frames becomes harder, increasing the need for finding relevant neighbouring users. The research suggests a flexible solution for dealing with the similarity limitation of current methods. To overcome the similarity problem, we suggest dividing items into multiple features. Using these features we extract several user interests, which can be compared among users.

4. Conclusion and Future scope

Collaboraive filtering is quite popular for the web-based applications and has also been proven to be successful when handling the web data. Yet, scope of further research is left as the data is highly dynamic in nature as well as its volume is increasing explosively.

Temporal information may be included in the form of time weightage function when calculating the similarity among the users-items in the process of CF. Hence this area can be explored to obtain an algorithm involving a new similarity function in CF to handle the dynamics of web data. This approach provides a solution to improve the overall CF performance. The dynamic nature of web data is captured as changes in the users' tastes or items' popularity are captured when new predictions are made. In the future, work can be extended on other algorithms such as multiclassifiers in the order to capture the time factors in the case of classification of web data using multiclassifiers. The time and space complexity can also be worked upon so as to provide better efficiency of this work. The behaviour of the web data being dynamic may also be analyzed as per other parameters so as to provide automated tools for its handling.

REFERENCES

[1]. Terveen, L., Hill, W., Amento, B., McDonald, D., and Creter, J. PHOAKS: *A System for Sharing Recommendations. Communications of the ACM*, 40(3). 1997, 59 62.

[2]. Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J. GroupLens: *Applying Collaborative Filtering to Usenet News. Communications of the ACM*, 40(3), 1997, 77-87.

[3]. Breese, J. S., Heckerman, D., and Kadie, C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 1998, 43-52.

[4]. Basu, C., Hirsh, H., and Cohen, W. Recommendation as Classification: Using Social and Content-based Information in Recommendation. *In Recommender System Workshop'98*, 1998, 11-15.

[5]. Ungar, L. H., and Foster, D. P. Clustering Methods for Collaborative Filtering. *In Workshop on Recommender Systems at the 15th National Conference on Artificial Intelligence.* 1998, 295

[6]. Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. *Analysis of Recommendation Algorithms for E-Commerce. In Proceedings of the ACM EC'00 Conference. Minneapolis, MN*,2000, 158-167

[7]. G.Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and management*, 24(5): 1988,513-523.

[8]. Nathan N Liu, Min Zhao, Evan Xiang, Qiang Yang, *Online Evolutionary Collaborative Filtering ACM*, 2010**,** 95-102**.**

[9]. G. Linden, B. Smith, and J. York, Amazon.com recommendations: item-to-item collaborative filtering, *IEEE Internet Computing, vol. 7, no. 1,*2003*,* 76–80.

[10]. G. Shani, D. Heckerman, and R. I. Brafman, An MDP-based recommender system, *Journal of Machine Learning Research*, vol. 6, 1265–1295,.

[11]. Lathia N., Hailes S., Capra L., Amatriain X., Temporal Diversity in Recommender Systems, *SIGIR* 2010.