Unison-CF: a multiple-component, adaptive collaborative filtering system

Manolis Vozalis and Konstantinos G. Margaritis

University of Macedonia, Dept. of Applied Informatics
Parallel Distributed Processing Laboratory
Egnatia 156, P.O. 1591, 54006, Thessaloniki, Greece
{mans,kmarg}@uom.gr, http://macedonia.uom.gr/~{mans,kmarg}

Abstract. In this paper we present the Unison-CF algorithm, which provides an efficient way to combine multiple collaborative filtering approaches, drawing advantages from each one of them. Each collaborative filtering approach is treated as a separate component, allowing the Unison-CF algorithm to be easily extended. We evaluate the Unison-CF algorithm by applying it on three existing filtering approaches: User-based Filtering, Item-based Filtering and Hybrid-CF. Adaptation is utilized and evaluated as part of the filtering approaches combination. Our experiments show that the Unison-CF algorithm generates promising results in improving the accuracy and coverage of the existing filtering algorithms.

keywords: collaborative filtering, memory-based filtering, adaptation, personalization, prediction, recommender systems

1 Introduction

Recommender Systems were introduced as a computer-based intelligent technique to assist with the problem of information and product overload. Their purpose is to provide efficient personalized solutions in e-business domains, benefiting both the customer and the merchant.

Two basic entities are featured in all Recommender Systems: the *user* and the *item*. A user utilizes the Recommender System, providing his opinion about items. The *goal* of the Recommender System is to generate suggestions about new items for that particular user. The process is based on the *input* provided, usually expressed in the form of ratings from that user, and the *filtering algorithm*, which is applied on that input. All the ratings provided by m users on n items are collected in a *m x n user-item matrix*.

Recommender algorithms can be roughly divided into two wide categories. *Memory-based* and *Model-based Systems*. Memory-based Systems are more efficient, in that they generate their recommendations without a need for any preprocessing. Nevertheless, they suffer from serious scalability problems. User-based Collaborative Filtering [1], and Content-based Filtering both belong to this category of filtering algorithms. A different approach is taken by Model-based Systems [2]. These algorithms, which often approach the problem from a probabilistic perspective [3], produce their predictions by first developing a model of user ratings. The construction of that model requires time but once created, the generation of the recommendations can be really fast.

Hybrid systems are based on the idea that an effective combination of different filtering techniques will improve the Recommender System's overall efficiency [4]. Among existing hybrids, such as Fab [5], Ripper [6], Filterbots [7], PTV [8], Condliff's two stage mixed-effects Bayesian Recommender [9], Content-Boosted Collaborative Filtering [10], and Hybrid-CF [11] we selected P-Tango as the basis for our filtering approach. Claypool et al. [12] proposed P-Tango as an approach which combines different filtering methods, by first relating each of them to a distinct component and then basing its predictions on the *weighted average* of the predictions generated by those components. Initially, they give equal weights to all ratings, but as more ratings are added, they adjust the weights so as to minimize past error. Still, they do not provide details about how this weight adjustment is achieved, while at the same time they limit their experiments to the combination of User-based and Content-based Filtering.

In this paper we present the Unison-CF algorithm, which refines and extends the work of Claypool et al [12]. Our approach in combining existing filtering algorithms is based on keeping them as separate components. Each algorithm is executed on its own, generating its predictions. The way that these predictions are combined, varies depending on the preferred implementation of the Unison-CF algorithm. By keeping the utilized filtering methods as separate components, we make sure that the Unison-CF Algorithm is extensible, since any new approach can be easily incorporated and contribute directly in the final prediction.

This paper can be outlined as follows. Sect. 2 provides information about the utilized data set and the evaluation metrics. Sect. 3 presents an overview of the three filtering algorithms involved in our Unison-CF experiments. Sect. 4 includes a detailed description of the general Unison-CF algorithm, while providing a formal discussion of two variations: The Basic and the Adaptive Unison-CF Algorithms. Sect. 5 presents a summary of experimental results and attempts an overall method comparison. The paper is concluded in Sect. 6.

2 Experimental Methodology

In order to execute the experiments described in the subsequent sections of this paper we utilized the data publicly available from the GroupLens movie recommender system. The MovieLens data set, used by several researchers [13] [14] [15], consists of 100.000 ratings, assigned by 943 users on 1682 movies. Users included have stated their opinions for at least 20 movies, while ratings follow the 1(bad)-5(excellent) numerical scale. That initial data set was used as the basis to generate five distinct splits into training and test data.

Many techniques have been proposed and used to evaluate Recommender Systems [15]. The choice among them should be based on the selected user tasks and the nature of the data sets. We wanted to derive a predicted score for already rated items rather than generate a top-N recommendation list. For that purpose we selected the two evaluation metrics to apply in our experiments. The first metric was *Mean Absolute Error (MAE)* [16]. It is a statistical accuracy metric which measures the deviation of predictions, generated by the Recommender System, from the true rating values, as they were specified by the user. The second metric utilized was *Coverage* [17]. It measures the percentage of items for which a filtering algorithm can generate predictions.

3 The Base Algorithms

In this section we discuss the three filtering algorithms which will be utilized by the proposed algorithm. More emphasis is put on the explanation of Hybrid-CF, a recently documented hybrid filtering approach, which combines elements from the two aforementioned filtering algorithms.

3.1 User-based Collaborative Filtering

The inspiration for User-based Collaborative Filtering methods comes from the fact that people who agreed in their subjective evaluation of past items are likely to agree again in the future [1]. The execution steps of the algorithm are (a) *Data Representation* of the ratings provided by m users on n items, (b) *Neighborhood Formation*, where the application of the selected similarity metric leads to the construction of the active user's neighborhood, and (c) *Prediction Generation*, where, based on this neighborhood, predictions for items rated by the active user are produced.

3.2 Item-based Collaborative Filtering

Item-based Filtering is based on the creation of neighborhoods. Yet, unlike the User-based Collaborative Filtering approach, those neighbors consist of similar items rather than similar users [18]. The execution steps of the algorithm are (a) *Data Representation* of the ratings provided by m users on n items, (b) *Neighborhood Formation*, where based on item similarities computed by the selected similarity metric, the active item's neighborhood is constructed, and (c) *Prediction Generation*, where predictions are calculated as a weighted sum of ratings given by a user on all items in the active item's neighborhood.

3.3 The Hybrid-CF Algorithm

Vozalis and Margaritis described a hybrid approach that combines elements from two basic recommendation algorithms - User and Item-based Collaborative Filtering [11]. The execution steps of the algorithm are (a) *Data Representation* of the ratings provided by m users on n items, (b) *Item Neighborhood Formation*, where a neighborhood of items most similar to the active item, that is the item for which we wish a prediction, is constructed, (c) *User Neighborhood Formation*, where we construct the active user's neighborhood based exclusively on items from the active item's neighborhood, and (d) Prediction Generation, where the users included in the active user's neighborhood are used to produce predictions of ratings by the active user. Steps (b) and (c) represent the respective contributions of User-based and Item-based Filtering in the recommendation procedure.

4 The Unison-CF Algorithm

The Unison-CF Algorithm *refines* past research work [12] by formally introducing the concept of "adaptation" in the recommendation process, which can be utilized in order to achieve weight adjustments. The Unison-CF Algorithm *extends* past research work (a) by increasing the number of filtering algorithms that participate in the experiments, and then, by comparing how different combinations of these methods, as components of the Unison-CF algorithm, contrast, and (b) by presenting an alternative implementation of the Unison-CF algorithm with fixed weights, and contrasting its results with cases where weight adjustments were implemented.

At this point, we will distinguish two alternative forms of the Unison-CF Algorithm: the Basic Unison Algorithm and the Adaptive Unison-CF Algorithm. In the Basic Unison-CF Algorithm, the participation of the filtering methods remains fixed and known from the beginning. On the other hand, the Adaptive Unison-CF Algorithm allows the errors, calculated throughout the recommendation process, to define the final participation of the filtering methods. The following sections discuss these two basic variations of the Unison-CF Algorithm. Each of them is supported by a number of specific algorithmic implementations and their experimental results.

4.1 The Basic Unison-CF Algorithm

The Basic Unison-CF Algorithm allows the contributing filtering algorithms, which from now on will be termed "base algorithms", to conclude with their prediction generation process. It, then, combines their generated predictions via a weighted sum of the following form:

$$uni\_pred_{aj} = w_1 * base\_pred_{1,aj} + w_2 * base\_pred_{2,aj} + ... + w_n * base\_pred_{n,aj} \quad (1)$$

This formula calculates the unison prediction of active user, u_a, on item i_j, based on the predictions generated by the n base algorithms, $base\_pred_{i,aj}$, with $i = 1, 2, ..., n$, and the weights, w_i, assigned to them. The final prediction is determined by two basic factors: a) the base algorithms that participate in the unison recommendation procedure, and b) the role assigned to each of these algorithms, mainly expressed through their weights..

In the implementations which will follow, we tested two distinct approaches regarding the role of the contributing filtering algorithms.

1. *Equal Contribution*: When more than one base algorithms can generate predictions for the same $\{$*user-item*$\}$ pair, the unison algorithm is set to assign identical significance to them. As a result, their contribution in the final recommendation stays fixed throughout the algorithm's execution, having an equal weight of $\frac{1}{n}$, where n is the number of contributing filtering algorithms. Consequently, Equal Contribution alters the general prediction formula of Basic Unison-CF Algorithm as follows:

$$uni\_pred_{ec,aj} = \frac{1}{n} * base\_pred_{1,aj} + \frac{1}{n} * base\_pred_{2,aj} + ... + \frac{1}{n} * base\_pred_{n,aj} \quad (2)$$

2. *Absolute Priority*: When more than one base algorithms can generate predictions for the same $\{$*user-item*$\}$ pair, the unison algorithm is set to always select the prediction by the *privileged* component, $base_{pr}$, and present it as the Recommender System's final prediction. Nevertheless, when the component given absolute priority cannot generate a prediction, the unison algorithm checks whether any other of the remaining base algorithms is able to fill that void. In that case, these components are utilized instead, boosting up the coverage. Consequently, Absolute Priority transforms the general prediction formula of Basic Unison-CF Algorithm as follows:

$$uni\_pred_{ap,aj} = \begin{cases} base\_pred_{pr,aj},\ if\ \exists base\_pred_{pr,aj} \\ w_2 * base\_pred_{2,aj} + ... + w_n * base\_pred_{n,aj},\ otherwise \end{cases} \tag{3}$$

   In the first case, when there exists a prediction from the privileged component, $\exists base\_pred_{pr,aj}$, we set $w_1 = 1$, while the contribution of all the other components is cancelled out by setting $w_i = 0$, for $i = 2, 3, ..., n$. In the second case, there is no prediction for the privileged component, $base\_pred_{pr,aj} = 0$, meaning that the overall unison prediction is defined by the rest of the contributing filtering algorithms by setting the values of their weights, w_i, for $i = 2, 3, ..., n$, according to the scheme we select. They can either be equal, which translates to Equal Contribution for all remaining components, or, if we stick with the Absolute Priority scheme, the next component generating a prediction, $base_k$, will be the only one utilized ($w_k = 1$), while all the subsequent components will be totally ignored ($w_i = 0$, for $i = k + 1, ..., n$).

4.2 The Adaptive Unison-CF Algorithm

The Basic Unison-CF Algorithm incorporated a pre-defined behavior, allowing the algorithm's execution steps to be fully anticipated from the beginning. The Adaptive Unison-CF Algorithm retains the general form of the prediction formula 1, which should be utilized to generate predictions for active user, u_a, on item i_j.

Still, the essence of the prediction formula is altered by the introduction of the concept of *adaptation*, which is reflected by the definition of the participating weights, w_i, for $i = 1, 2, ..., n$. The basic idea behind adaptation is that the algorithm's behavior will change, or more precisely *adapt*, during the recommendation process. At this point, we can distinguish two diverse approaches of applying adaptation in the unison algorithm:

1. *Adaptation on a single, preceding user*: According to the first approach, the recommender system's behavior adapts by basing its predictions for the *current* user solely on the prediction errors observed for the *previous* user. Specifically, before proceeding with predictions for user u_a, we collect the predictions that the base algorithms, $base_1, base_2, ..., base_n$, generated for user u_{a-1}, and calculate their cumulative accuracy errors, $E_{1,a-1}, E_{2,a-1}, ..., E_{n,a-1}$. Error $E_{c,a-1}$ with $c = 1, 2, ..., n$, corresponds to the sum of differences between the predictions, $pr_{c,a-1j}$, generated by component $base_c$ for all items, $j = 1, 2, ..., l$, rated by pre-

vious user u_{a-1}, and the user's actual ratings, r_{a-1j}, on those items:

$$E_{c,a-1} = \sum_{j=1}^{l} |pr_{c,a-1j} - r_{a-1j}| \tag{4}$$

Still, a filtering component may not be able to generate predictions for all l items that have been rated by user u_{a-1}. Let's assume that the number of items rated by user u_{a-1} for which base algorithms $base_1, base_2, ..., base_n$ were able to generate predictions are $l_{1,a-1}, l_{2,a-1}, ..., l_{n,a-1}$, where obviously $l_{c,a-1} \leq l$, for $c = 1, 2, ..., n$. Bearing that in mind and starting from the cumulative accuracy errors, $E_{c,a-1}$, we can now proceed and calculate the accuracy errors per prediction, $E_{pr,c,a-1}$, as follows:

$$E_{pr,c,a-1} = \frac{E_{c,a-1}}{l_{c,a-1}} \quad for\ c = 1, 2, ..., n \tag{5}$$

Error $E_{pr,c,a-1}$ represents the error per prediction generated by filtering component $base_c$ for user u_{a-1}. Taking these errors into consideration, we assign different weights, w_c for $c = 1, 2, ..., n$, to the base filtering approaches. These weights, normalized so that they add up to unity, are defined to be proportionally inverse to the same component's recommendation error per prediction for the previous user, $E_{pr,c,a-1}$:

$$w_c = \frac{1}{E_{pr,c,a-1}} \tag{6}$$

Once specified, the weights can be utilized in the unison prediction formula 1. As a result, a method with low prediction error for the previous user will have a higher impact in the prediction for the succeeding user. At the same time, a component with increased prediction error for the preceding user will assume a diminished role in the same prediction generation.

2. *Adaptation on all preceding users*: According to the second approach, the contribution of the base algorithms in the final recommendation adapts not by taking into account the predictions generated for a single, preceding user, but by collecting the predictions produced by the participating components for *all* preceding users, that is for users u_1 to u_{a-1}. Based on these predictions we evaluate the cumulative accuracy errors, $E_{c,all}$, for each of the n components and, thus, their participation in the total accuracy error. Error $E_{c,all}$ with $c = 1, 2, ..., n$, is calculated as follows:

$$E_{c,all} = \sum_{u=1}^{a-1} \sum_{j=1}^{l} |pr_{i,uj} - r_{uj}| \tag{7}$$

According to the formula, we start by computing the sum of differences between the predictions, $pr_{c,uj}$, generated by component $base_c$ for all items, $j = 1, 2, ..., l$, rated by user u, and the user's actual ratings, r_{uj}, on those items. We proceed by adding these sums for all users from u_1 to u_{a-1}. We can now calculate error $E_{pr,c,all}$, that is the error per prediction generated by filtering component c for all preceding users, by executing the steps which were described in the previous paragraph. These errors, computed for each filtering component, are added up in order to obtain the overall components error. The weights assigned to each component are set to be inverse proportionally to the corresponding component's participation in the total accuracy error. The difference from the previous approach is that by taking into account the whole prediction history, and not just the last user, the weight adaptation becomes slower as the recommendation procedure proceeds and more users are considered.

5 Overall comparison of Unison-CF implementations with base cases

In this section we will report the results from our experiments with seven distinct implementations of the Unison-CF algorithm, where we varied the number and type of filtering algorithms involved, as well as the way those filtering algorithms were combined. The main attributes of the Unison-CF implementations are outlined in Table 1.

Table 1. Brief Description of Unison-CF Implementations tested

	components	**adaptability**	**functionality**
Unison1	ub, ib	no	priority to the ub component
Unison2	ub, ib	no	equal weights to both components
Unison3	ub, ib, hcf	no	equal weights to all 3 components
Unison4	ub, ib, hcf	yes	adaptation on single, preceding user
Unison5	ub, ib, hcf	yes	adaptation on all previous users
Unison6	ub, hcf	yes	adaptation on all previous users
Unison7	ub, hcf	yes	adaptation on single, preceding user

Fig. 1 compares the mean absolute error and coverage values for the Unison-CF implementations we tested. It also contrasts them with the corresponding MAE and coverage values from User-based (*user-b*), Item-based (*item-b*) Filtering and Hybrid-CF (*hyb-cf*), in order to see how the Unison-CF implementations relate with the filtering algorithms that they combine.

A careful review of the coverage and accuracy figures leads to the following conclusions.

- The combination of different filtering approaches in a single Unison-CF implementation cannot improve on the best MAE values achieved by any of the involved filtering approaches. Its accuracy will lie between the best and the worst case of the filtering approaches that it unites.
- Unison-CF implementations are followed by a considerable increase in the coverage, when compared to the coverage of the filtering algorithms that they unite. Any increase in the number of components of a Unison-CF implementation may lead to a further increase of the average coverage, as documented by the coverage values of

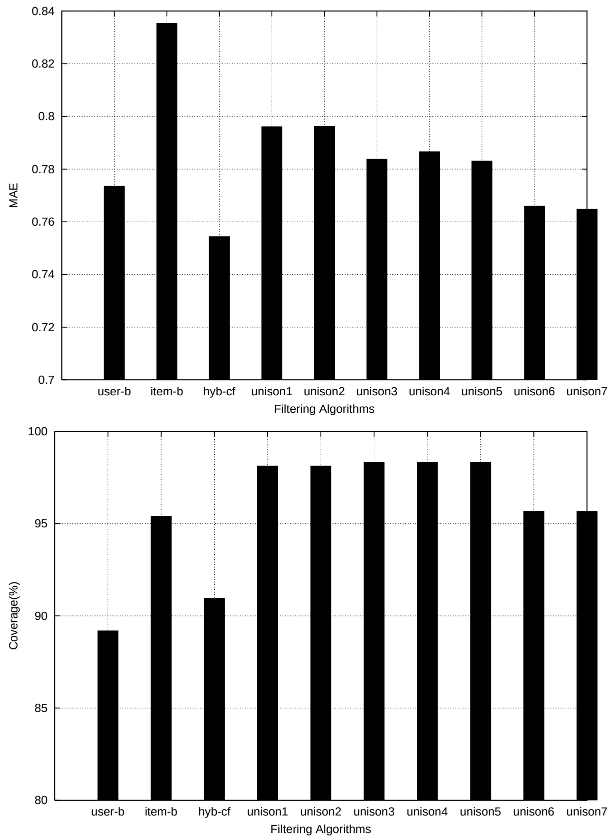

Fig. 1. Overall comparison of Unison-CF implementations with base cases

Unison-CF implementations that unite 3 base algorithms, when compared to those uniting only 2.

- If we want to single out the filtering algorithm with the best results regarding *accuracy*, we have to select Hybrid-CF. Its average MAE of 0,7545 was the best, with Unison7 (average MAE=0,7649) representing the next best case.
- If we want to single out the filtering algorithm with the best results regarding *coverage*, we have to select Unison5. Its average coverage of 98,3413% was the same as the coverage values achieved by Unison3 and Unison4. Nevertheless, Unison5 displayed a slightly better behavior as far as accuracy was concerned, and as a result it represents the best choice.
- If we want to single out the filtering algorithm with the best overall behavior, by taking into account both accuracy and coverage, we have to select Unison7. Its average MAE of 0,7649 was the second lowest, trailing only Hybrid-CF, when the rest of the Unison-CF implementations had MAE values ranging from 0,7963, as in the case of Unison2, to 0,7660, as in the case of Unison6. This significant advantage of Unison6 in accuracy did justify an average coverage of 95,6888%, which was

slightly worse when compared to the average coverage values of the rest of the Unison-CF implementations, ranging around 98%.
– Adaptation on a single user fared extremely well against adaptation on all users. This result was unexpected and requires further investigation.

6 Conclusions

In this work we have presented the Unison-CF Algorithm, a hybrid filtering technique which can combine two or more collaborative filtering approaches with the help of a weighted sum. We also discussed two distinct variations of the Unison-CF Algorithm: The Basic Unison-CF Algorithm and the Adaptive Unison-CF Algorithm. The former incorporates a pre-defined behavior, expressed by the fixed values of its weights. The latter introduces the concept of adaptation, allowing the weights to adjust during the recommendation process. All discussions regarding the aforementioned algorithms did include a formal and detailed description of the execution steps required, therefore allowing them to be easily extended or applied on different base algorithms than those tested.

The descriptions of the proposed algorithms were supported by numerous experimental implementations, each of them incorporating distinct parameter settings. The settings were selected in order to contrast the utility of the Unison-CF Algorithm, in both of its variations, with the filtering approaches that it unites. The experimental results proved the promise held by the utilization of adaptation in the recommendation process. Furthermore, specific Unison-CF implementations displayed an overall behavior which improved on the base algorithms, according to the applied evaluation metrics.

For our future work we plan on experimenting with the concept of an ”adaptation window”, in order to test how a changing number of users participating in the Adaptive Unison-CF Algorithm will affect its efficiency.

References

1. Resnick, P., Iacovou, N., Sushak, M., Bergstrom, P., Riedl, J.: Grouplens: An open architecture for collaborative filtering of netnews. In: ACM 1994 Conference on Computer Supported Cooperative Work, New York, NY (1994) 175–186
2. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI (1998)
3. Chen, Y.H., George, E.I.: A bayesian model for collaborative filtering. In: Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics. (1999)
4. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction **12** (2002) 331–370
5. Balabanovic, M., Shoham, Y.: Fab: Content-based, collaborative recommendation. Communications of the ACM **40** (1997)
6. Basu, C., Hirsh, H., Cohen, W.: Recommendation as classification: Using social and content-based information in recommendation. In: Proceedings of the 15th National Conference on Artificial Intelligence, Madison, WI (1998)

7. Sarwar, B.M., Konstan, J.A., Borchers, A., Herlocker, J., Miller, B., Riedl, J.T.: Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In: Conference on Computer Supported Cooperative Work. (1998)
8. Smyth, B., Cotter, P.: Surfing the digital wave: Generation personalized tv listings using collaborative, case-based recommendation. In: Third International Conferece on Case-based Reasoning, Munich, Germany (1999)
9. Condliff, M.K., Lewis, D.D., Madigan, D., Posse, C.: Bayesian mixed-effects models for recommender systems. In: ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation, Berkeley, CA (1999)
10. Melville, P., Mooney, R.J., Nagarajan, R.: Content-boosted collaborative filtering. In: ACM SIGIR Workshop on Recommender Systems, New Orleans, LA (2001)
11. Vozalis, E., Margaritis, K.G.: On the combination of user-based and item-based collaborative filtering. Technical report, University of Macedonia, Greece (2003)
12. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M.: Combining content-based and collaborative filters in an online newspaper. In: ACM SIGIR Workshop on Recommender Systems-Implementation and Evaluation, Berkeley, CA (1999)
13. Schein, A.I., Popescul, A., Ungar, L.H., Pennock, D.M.: Methods and metrics for cold-start recommendations. In: ACM SIGIR-2002, Tampere, Finland (2002)
14. Ujjin, S., Bentley, P.J.: Particle swarm optimization recommender system. In: Proceedings of the IEEE Swarm Intelligence Sympoisum 2003, Indianapolis (2003)
15. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems **22** (2004) 5–53
16. Shardanand, U., Maes, P.: Social information filtering: Algorithms for automating 'word of mouth'. In: Proceedings of Computer Human Interaction. (1995) 210–217
17. Herlocker, J.L.: Understanding and Improving Automated Collaborative Filtering Systems. PhD thesis, University of Minnesota (2000)
18. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.T.: Item-based collaborative filtering recommendation algorithms. In: 10th International World Wide Web Conference (WWW10), Hong Kong (2001)