

Locally Adaptive Neighborhood Selection for Collaborative Filtering Recommendations

Linus Baltrunas and Francesco Ricci

Free University of Bozen-Bolzano,
Domenikanerplatz 3,
Bozen, Italy

Abstract. User-to-user similarity is a fundamental component of Collaborative Filtering (CF) recommender systems. In user-to-user similarity the ratings assigned by two users to a set of items are pairwise compared and averaged (correlation). In this paper we make user-to-user similarity adaptive, i.e., we dynamically change the computation depending on the profiles of the compared users and the item whose rating prediction is sought (target item). We propose to base the similarity between two users only on the subset of co-rated items which best describes users' tastes with respect to the target item. These are the items which have the highest correlation with the target item. We have evaluated the proposed method using a range of error measures and showed that the proposed locally adaptive neighbor selection, via item selection, can significantly improve the recommendation accuracy compared to standard CF.

1 Introduction

The World Wide Web, interconnecting a myriad of information and business services, has made available to on-line users an over abundance of information and very large product catalogues. Hence, users trying to decide what information to consult or what products to choose may be overwhelmed by the number of accessible options. Collaborative Filtering (CF) is a recommendation technique which emulates a simple and effective social strategy called “word-of-mouth” and is now largely applied in Web 2.0 platforms. In CF personalized recommendations for a target user are generated using opinions, i.e., ratings on items, of users having similar tastes to that of the target user [1]. A CF system represents users with their ratings on a set of items (ratings vectors). When requested to generate a recommendation for a target user, a (memory based) CF system first selects a set of similar users according to a similarity measure computed on their ratings vectors (neighborhood selection). Then, it generates rating predictions for items not rated yet by the target user. Finally the system recommends the items with the highest predicted rating.

User-to-user similarity plays a very important role in CF; it is used in the neighborhood selection and in the final rating prediction, and it is normally computed using all the items co-rated by the two users whose similarity is required.

The work presented in this paper is aimed at improving CF by adapting the user-to-user similarity function used in the neighbor selection step taking into account: a) the user for whom a rating prediction is sought (active user), and b) the particular target item for which the system is predicting the rating of the active user. We hypothesized that the neighbor, and consequently the goodness of the rating prediction, can be improved if the user-to-user similarity is based on a well selected subset of commonly co-rated items: those items highly correlated with the target item. The rationale is that information about some items may not be relevant, and even detrimental when predicting the rating. For example, if we try to predict the rating for the movie “The Matrix”, it could be better to take into account ratings on similar action or sci-fi movies and ignore ratings on documentary movies.

In addition to the improvements in the precision we shall show that the proposed adaptive user-to-user similarity method increases the diversity of the subsets of profiles of the neighborhood users yielding a better recall. The rationale is that a target user may be similar to two users because highly correlated to them on two different subsets of items. Hence these two users are more likely to cover different parts of the item space. In fact, the proposed locally adaptive neighbor selection method brings improvements for all the error measures we used.

The paper is organized as follows. Section 2 discusses the related work on the item selection. Section 3 describes item selection methods for CF and in Subsection 3.1 we describe the item weighting methods that were used to determine importance of the item. The proposed locally adaptive item selection method is evaluated in Section 4. Finally, Section 5 draws the conclusions and presents future work.

2 Related Work

Item selection techniques are well know tools used in Machine Learning (ML). In ML they are called feature selection and are widely used to improve the prediction accuracy of supervised classification [2, 3]. Recently they are receiving a new interest because of their exploitation in Information Retrieval methods based on learning [4, 5]. In fact, a user-based CF system can be described as a collection of instance-based classifiers, one for each item, whose’ rating is sought. Given a target item (class) and an active user, the user ratings on all other items provide the instance description (predictive items). In this perspective, the rating prediction step of a CF system can be described as a classification or regression learning problem, i.e., one classification/regression problem for each target item’s rating prediction. The similarity measure is based on users’ preferences, i.e., item ratings. Hence, these items can be regarded as user’s features, and item ratings are the values of the feature. In general, the huge search space of thousands and millions of items makes the available feature selection methods hard to apply to CF. In fact, to our best knowledge item selection has not been explored in the context of the CF.

Classical feature selection methods assume that features are either relevant in the whole instance space or irrelevant throughout. However, it can often be the case that features are relevant only in the context of other features. To address this situation local feature selection was studied in [6, 7]. Here, features are selected depending on the target instance whose class must be predicted, and the other features which are present in the selection set. Both works are related to our approach, as in these cases the selected features depend on the target instance (user in a CF setting). However, they study and exploit the relationships between predictive features rather than the dependency between predictive items (features) and target item (feature), as we do.

Item weighting for CF is another related approach, which tries to adapt user-to-user similarity depending on the prediction task and the profiles of the users [8–10]. Here, the items with larger weights will have a major influence in the user-to-user similarity computation. Due to the huge amount of items, sparse data and non localized approaches all item weighting methods, not shown here for lack of space, give only minor improvements over the classical CF.

3 Item Selection for Collaborative Filtering

Finding the optimal subset of items would require to conduct an extensive search in the space of all the subsets of the items [3]. Applying this to a recommender system scenario would require to conduct a search procedure for every target item (the item playing the role of the class to be predicted) and for a large number of subsets of the predictive items. This is clearly extremely expensive, and therefore, we propose to use a more parsimonious approach (filter method) that uses information provided by a item weighting method to select, for each target item and user pair, an appropriate set of predictive items.

Hence, first we compute item weights, using one of the methods described later in Section 3.1, and then we filter out irrelevant items, i.e., those with the smallest weights, for a given target item rating prediction. Following this procedure, every item weighting method generates one or more corresponding item selection method depending on how the weights are used to select items.

Extremely sparse rating matrix makes classical filter method ineffective. Imagine for instance, that there are two users that are perfectly correlated but have co-rated just a few items and these items are not (in general) strongly correlated with the target item. One of the two users could be used to predict ratings of second user. But, if we select a small number of items, according to a measure of item-to-item correlation, we have a very small chance that these users will overlap on the selected items. Therefore, using classical filter method many good neighbors could be discarded hence decreasing the prediction accuracy.

Instead we proposed a localized item selection method called **BIPO** (Best Items per Overlap). It selects subset of items with biggest weights from the set of items co-rated by both users whose similarity we want to determine. We shall explain BIPO method using a simple example of user-item rating matrix showed below:

	i_1	i_2	i_3	i_4	i_5	i_t
w_{ti}	0.1	0.2	0.3	0.4	0.5	
u_1	5	3	2	1	?	6
u_2	4	2	4	?	5	?

The table consist of two user and six items. The question marks indicate the unknown ratings. Let us assume that we want to predict user's u_2 rating for the item i_t . Moreover, suppose that we have computed item weights beforehand, using an item weighting algorithm. The weights are showed in the second line of the table. For example, the weight of item i_1 for predicting the target item i_t is $w_{t1} = 0.1$.

In the example above, BIPO method would select items with the highest weights that are rated by both users. Suppose we want to compute user-to-user similarity on the 2 most correlated items. In such a case BIPO would select items 3 and 2, whereas, all other items would be not considered, despite the fact that they could have bigger correlation with a target item. Hence, in BIPO the items used in each prediction change for every target item and neighbor user pairs.

3.1 Weight Computation Methods

We use item weighting to estimate how much a particular item is important for deciding about the class membership of instances. Only the items with the highest weights are selected using BIPO to compute the user-to-user similarity. In CF, item weights can be learned while exploring training data consisting of user ratings, or using external information associated with the items. In this paper we introduce five item weighting methods used for item selection: Random, Variance, Mutual Information, Tag label based, and IPCC.

Random. The first method is used only as the baseline for comparisons and uses a random item weighting. Random weights in $[0, 1]$ are selected for each target and predictive items combinations.

Variance. Variance method was originally proposed by [11] and gives higher weight to the items with higher variance among the ratings provided by the users to that item:

$$w_{ji}^V = w_i^V = \frac{\sum_{u \in U(i)} (v_{ui} - \bar{v}_i)^2}{|U(i)|}$$

here \bar{v}_i is the mean of the ratings of item i , $U(i)$ is the set of users who rated item i , and $|U(i)|$ is the number of users who rated item i . Variance weighting method uses only information about the predictive item and does not take into account the target item for which a prediction is sought.

IPCC. The method computes the weights using the Pearson Correlation Coefficient (PCC) between two item rating vectors, i.e., the ratings of all users for the two items, as a measure of dependency between the two items.

$$w_{ji}^P = \frac{\sum_u (v_{ui} - \bar{v}_i)(v_{uj} - \bar{v}_j)}{\sqrt{\sum_u (v_{ui} - \bar{v}_i)^2 \sum_u (v_{uj} - \bar{v}_j)^2}}$$

here u runs over all the users that have rated both i and j , and \bar{v}_i is the mean of item i ratings.

Mutual Information. Mutual Information measures the information that a random variable provides to the knowledge of an other. Mutual Information between two items is defined as :

$$w_{ji}^M = \sum_{x \in j} \sum_{y \in i} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

and in practice it is implemented using entropy as in [8].

Tag weighting. The previous methods exploit statistics of the users' rating data to compute the item weights. The last method we present here computes the weights using description of the items. In the movie recommendation data set, which we are going to use for our experiments, movies are tagged with movie genres. Hence, we make the assumption that the larger the number of common tags (genres), the higher is the dependency. The weight of the predictive item i for a prediction of the ratings of item j is given by:

$$w_{ji}^T = \frac{\# \text{ comon tags of } i \text{ and } j}{\# \text{ tags}}$$

Tag-based item weighting is related to the methods presented in [12] where item description information is used to selectively choose the items to be used in the user-to-user correlation.

For efficiency reasons, for all the item selection algorithms, we off-line compute all the weights and later we use these stored values in the user-to-user similarity computation. In practice, to store all the weights, we need an $M \times M$ matrix of weights, where M is the cardinality of the item set. In other words, one vector of weights of size M is used for each item prediction.

4 Experimental Evaluation

In this section we present the evaluation of the BIPO item selection method for neighbor selection. As we already mentioned above, our method is computing the set of item weights and later uses them to select items considered in the user-to-user similarity. We note that PCC between two users is computed on the subset on selected items **only** while computing the neighborhood of the active user, whereas the standard PCC, without item selection, is used to compute the predicted rating:

$$v_{xj}^* = \bar{v}_x + \frac{\sum_{y \in N(k, x, j)} PCC(x, y) \times (v_{yj} - \bar{v}_y)}{\sum_{y \in N(k, x, j)} |PCC(x, y)|}$$

here the sum runs on the k -nearest neighbors of the user x , $N(k, x, j)$, which depends on target item j . In our implementation, as previously done in other studies [12], we do not take into account neighbors which have less than six co-rated items.

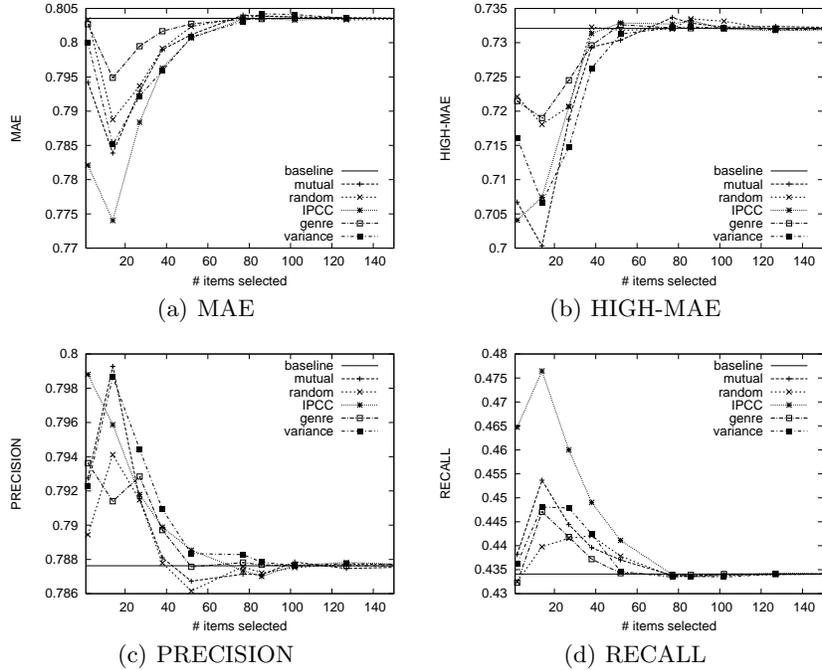


Fig. 1. Performance of BIPO item selection methods for different error measures.

For our evaluation we used MovieLens [13] dataset with ratings in $\{1, 2, 3, 4, 5\}$. It contains 100K ratings, for 1682 movies by 943 users, who have rated 20 and more items. The data sparsity is 96%. To evaluate the proposed methods the data set was randomly divided into train (80%) and test (20%) subsets. We used the train data to learn the weights and also to generate a prediction for the test ratings. We evaluated the performance of the proposed methods with a wide range of error measures.

To measure the accuracy we used: MAE, High MAE, F measure, precision and recall [11, 14]. To compute F, precision and recall, we considered items worth recommending (relevant items) only if their ratings were 4 or 5. Since, we are interested in recommending only the top items, we propose to modify the MAE error measure to see how an algorithm performs on the predictions of items with highest ratings. For this purpose we defined High MAE measure as MAE computed only on the items that were rated by the user 4 or 5. In the experimental part we focus on the behavior of BIPO item selection. To compute the weights we used five item weighting methods described in section 3.1. In figure 1 the performance of all these methods for the MovieLens data set are depicted. We note that **all** item weighting methods used for BIPO item selection showed a better performance over the baseline CF ,i.e., collaborative filtering without item selection, for all the error measures used. This result is important since it clearly

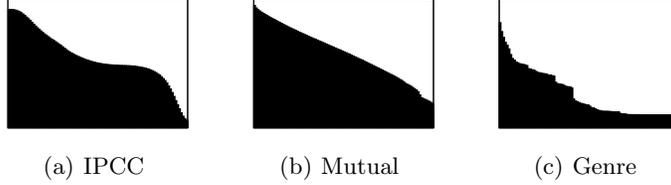


Fig. 2. Various correlation distributions.

shows the robustness of item selection, and clearly shows that the improvements are due to item selection rather than to weighting.

In fact, there is no single best item weighting approach and the winner depends on the particular error measure used. For example, weighting based on Mutual Information (in fig. “mutual”) produces an item selection that performs best for High-MAE error measure, whereas IPCC weighting performs better for recall measure and gives improvement up to 6.6%. Random item selection and genre labeling methods are the worst, however, they also improve the performance of the baseline method. Note that for precision, MAE, and High MAE the Variance based method performs as good as IPCC and Mutual Information based methods. This is important, because in Variance based approach the weight of the item does not depend on the target item, therefore, this method can be easily applied for large data sets with many items. It can be also efficiently computed and cached.

Similar performances can be seen using other datasets. Due to the lack of space, we could not include the full description of result in this paper. The method was tested on Yahoo! WebScope dataset[15], which contains extremely sparse data. We discovered, that the improvements are smaller, however, they increase when the average overlap between users increases (achieved by filtering out users with small number of ratings). Our conclusion is that in order to make a meaningful item selection we must have enough co-rated items to choose from. In order to explain why some weighting methods perform better than others we analyzed the distribution of correlation values between items. We ordered item weights for each of the target items in the descending order and took average value for each of the position. In such a way we got an ordered list of M elements which represents $M \times M$ weight matrix. In order to be able to depict this list we further compressed this representation into a histogram with 100 bins, where each bin averages 1% of ordered list values. In Figure 2 three different distributions of item-to-item correlation measures for MovieLens dataset are depicted.

It is important to note that we are not interested in the absolute values of the weights, but rather in their relative size. Given the target item, we select the top f items for the similarity computation. The problem arises when we have a lot of weights with almost equal values. In such a case the items will be ordered in a pseudo random order, leading to a wrong item selection. Such a bad case of weighting can be seen for Genre weighting (see Figure 2(c)), where

flat parts represents items with the same weigh value. This explain the bad performance of genre item weighting (see Figure 1). The other two weighting methods produce relatively good ordering, i.e., there is a small number of items with similar weights.

In the previous experiments we showed that BIPO item selection method gives good results using a small number of items in user-to-user similarity. In the next analysis we studied if BIPO is capable to generate predictions for the same user-item pairs that standard CF will predict¹. Hence we evaluated the performance of BIPO on the subsets of the items whose rating is predicted by both BIPO and CF, and items that where predicted only by one of the two methods.

The performance of BIPO item selection method with fixed number (14) of selected items was compared against the baseline CF. Both methods used 60 k-nearest neighbors. The test set was divided into three subsets: “common” subset contains items predicted by both methods; *CF – only* contains the items whose ratings were predicted only by the baseline CF and *BIPO – only* contains items that were predicted only by BIPO. In table 1 summary of the results is shown.

Table 1. Performance of BIPO and standard CF on different set of ratings.

Measure	common ratings		CF-only ratings	BIPO-only ratings
	CF	BIPO	CF	BIPO
# rat. predicted	17833	17833	117	1146
MAE	0.8024	0.7663	0.9656	0.8939
HIGH-MAE	0.7305	0.6954	1.1055	0.9776

These results show that both methods can predict a large common subset user-item pairs in the test data set, namely 17833. On this set of user-item pairs that both can predict, BIPO decreases MAE by 4.5%. Moreover, BIPO can make more predictions compared to the baseline method (117 vs 1146 items not in the common subset). This result is surprising and shows that despite data sparsity, carefully selecting items for neighborhood formation increases recall and could also improve recommendation diversity.

The increase in recall (and also in the ability to make more predictions) shown by BIPO can be explained by better analyzing the way user-to-user similarity is computed by this method. Firstly of all, BIPO computes the similarity using only the items, which have highest correlation with the target item. This can explain why it is more likely that the target item is also rated by the neighbors found by this method, and therefore the collaborative filtering prediction rule can actually compute a prediction for it (this is an obvious precondition).

Secondly, by computing the similarity on a smaller subset of items we tackle a problem related to (target) users, who rated many items. At the first glance, such users should be easier to serve, and will be provided with better recommendations, because they have a large rating history. However, when looking for

¹ Thanks to J. Konstan for suggesting this study in a personal communication.

neighbors it is likely that there will be many users highly correlated to this target user but only on a small number of overlapping items. At the end, without item selection, we could select neighbors who are highly correlated but on a small number of items. Using item selection we compute overlap only on a small but **highly selected** items, making more likely that other users with larger profiles become neighbors of the target user. This may decrease the average correlation in a user neighborhood but computes correlations with higher reliability.

Moreover, since the selection of items is made by analyzing both the target user and the neighbor profile, the neighbors found are more diverse, as they can be correlated on different subsets of items. This is very important and explain the increase in recall as the increased diversity of the neighbors increases the chance that some neighbors rated the target item.

5 Discussion and Conclusions

In this paper we introduced locally adaptive item selection method for an improved neighbor selection in CF. We evaluated the proposed method along with a range of error measures and varying the number of nearest neighbors. In summary, BIPO item selection approach, using a small number of items, can achieve significant improvements for all the considered error measures. This result is important because it shows that CF user profiles contain a lot of redundancy and even if the data set is sparse the information is not uniformly distributed among users.

We showed that CF performances can be improved with a careful selection of the item ratings, i.e., acquiring ratings for certain items can greatly improve the performance of the recommendation algorithm. In fact, since BIPO select items according to the target item, the outcome can be a different neighborhood for the same target user depending on the prediction that must be computed. Hence BIPO is in fact an item selection and user selection method. The idea is that when we must make a prediction for a particular type of item, not all the neighbors computed by the classical CF may be relevant. If a neighbor is highly correlated to the target, but on items significantly decorrelated with the target item, then this should not be considered.

We also shown that in item selection the precise weighting method is not crucial. In other words, as we use only ranking of the item weights, but not the absolute values, even a suboptimal weighting can generate a good item selection. This observation is the base for a future extension of this work where we plan to consider fast sub-optimal weighting schemes for large data sets. We stress again that we used item selection only in the neighborhood formation step, therefore, we state that item selection leads to a better neighborhood formation. A natural extension of the method would be to validate item selection also when used in the prediction step.

In the future we shall better analyze the user selection problem, in addition and in combination with item selection. We want to compare pure item weighting and selection methods with instance (user) selection methods. The computation

of the item weights is an expensive step. We need to recompute the weights when new ratings are registered to the system. Therefore, for this reason we are currently working on the design of a item weighting method based on the RELIEF [16] feature estimator in order to avoid item weights recomputation. Using this method with every new rating we could gradually adapt the overall weighting schema.

References

1. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: *The Adaptive Web*, Springer Berlin / Heidelberg (2007) 291–324
2. Wettschereck, D., Aha, D.W., Mohri, T.: A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artif. Intell. Rev.* **11**(1-5) (1997) 273–314
3. Kohavi, R., John, G.H.: Wrappers for feature subset selection. *Artificial Intelligence* **97**(1-2) (1997) 273–324
4. Radlinski, F., Joachims, T.: Query chains: learning to rank from implicit feedback. In: *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, New York, NY, USA, ACM Press (2005) 239–248
5. Geng, X., Liu, T.Y., Qin, T., Li, H.: Feature selection for ranking. In: *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, ACM (2007) 407–414
6. Chris Atkeson, Andrew Moore, S.S.: Locally weighted learning. *AI Review* **11** (April 1997) 11–73
7. Puuronen, S., Tsymbal, A.: Local feature selection with dynamic integration of classifiers (2001)
8. Yu, K., Xu, X., Ester, M., Kriegel, H.P.: Feature weighting and instance selection for collaborative filtering: An information-theoretic approach*. *Knowl. Inf. Syst.* **5**(2) (2003) 201–224
9. Jin, R., Chai, J.Y., Si, L.: An automatic weighting scheme for collaborative filtering. In: *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, ACM Press (2004) 337–344
10. Breese, J.S., Heckerman, D., Kadie, C.M.: Empirical analysis of predictive algorithms for collaborative filtering. In Cooper, G.F., Moral, S., eds.: *UAI*, Morgan Kaufmann (1998) 43–52
11. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: *SIGIR*, ACM (1999) 230–237
12. Berkovsky, S., Kuflik, T., Ricci, F.: Cross-domain mediation in collaborative filtering. In Conati, C., McCoy, K.F., Paliouras, G., eds.: *User Modeling*. Volume 4511 of *Lecture Notes in Computer Science.*, Springer (2007) 355–359
13. MovieLens: Movielens dataset, <http://www.grouplens.org/>
14. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **22**(1) (2004) 5–53
15. Yahoo!: Webscope movie data set. <http://research.yahoo.com/> (Version 1.0)
16. Robnik-Šikonja, M., Kononenko, I.: Theoretical and empirical analysis of relief and relief. *Mach. Learn.* **53**(1-2) (2003) 23–69