# A Hybrid Approach for Improving Prediction Coverage of Collaborative Filtering

Manolis G. Vozalis, Angelos I. Markos and Konstantinos G. Margaritis

**Abstract** In this paper we present a hybrid filtering algorithm that attempts to deal with low prediction Coverage, a problem especially present in sparse datasets. We focus on Item HyCoV, an implementation of the proposed approach that incorporates an additional User-based step to the base Item-based algorithm, in order to take into account the possible contribution of users similar to the active user. A series of experiments were executed, aiming to evaluate the proposed approach in terms of Coverage and Accuracy. The results show that Item HyCov significantly improves both performance measures, requiring no additional data and minimal modification of existing filtering systems.

## 1 Introduction

Recommendations are generated based on taste information from users with similar interests in common items. Recommender Systems (RS), described as computer-based intelligent techniques which can provide personalized recommendations, were introduced to alleviate the problem of information and product overload.

RSs utilize various types of data and tools in order to achieve their purpose. Collaborative Filtering (CF) is one of the most successful methods among those utilized by RSs. It applies Information Retrieval and Data Mining techniques to extract automated recommendations for a user, based upon the assumption that users who have agreed in the past, tend to agree in the future.

A number of fundamental problems may reduce the quality of the predictions generated by a RS. Among others we have to mention *sparsity*, which refers to the problem of insufficient data, *scalability*, which refers to a performance degradation following a possible increase in the amount of data involved, and *synonymy*, which is

Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece,
e-mail: {man,amarkos,kmarg}@uom.gr

caused by the fact that similar items may have different names and cannot be easily associated. Furthermore, a filtering algorithm should be able to generate accurate predictions, as measured by the appropriate metric of choice.

The fraction of items for which predictions can be formed over the total number of rated items, is measured by prediction Coverage [4]. It is a common occurrence that a RS will not be able to provide a prediction for specific users on specific items because of either the sparsity in the data or other parameter restrictions, which are set during the system's execution. Systems with low Coverage are less valuable to users, since they will be limited in the decisions they are able to help with. On the other hand, RSs with high Coverage, combined with a good accuracy measure, will correspond better to user needs.

Hybrid systems [3] combine different filtering techniques in order to produce improved recommendations. Content-Boosted Collaborative Filtering [6] improves on user-based predictions by enhancing the initial matrix of ratings through the application of a content-based predictor. Wang et al. [12] formulate a generative probabilistic framework and merge user-based predictions, item-based predictions and predictions based on data from other but similar users rating other but similar items. Jin et al. [5] propose a Web recommendation system which integrates collaborative and content features under the maximum entropy principle.

In this paper, we present a hybrid approach that increases the percentage of items for which predictions can be generated, while it can potentially improve the system's accuracy. The proposed algorithm combines Item-based and User-based CF implementations in an attempt to effectively deal with the prediction coverage problem. A series of experiments were executed in order to evaluate the performance of the proposed approach.

The rest of this paper is organized as follows: In Section 2 we describe in brief two existing CF algorithms we built our work upon. In Section 3 we sketch the outline of Item HyCov, a hybrid approach. The results of two different sets of experiments that compare the proposed approach with Item-based filtering are discussed in Section 4. Finally, in Section 5 we draw the conclusions from the outcome of our experiments and present the future work.

## 2 The Base Algorithms

In this section we discuss in brief the two filtering algorithms which will be utilized by the proposed approach, *User-based Collaborative Filtering* (UbCF) and *Item-based Collaborative Filtering* (IbCF).

The inspiration for UbCF methods comes from the fact that people who agreed in their subjective evaluation of past items are likely to agree again in the future [7]. 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 neigh-

borhood, and (c) *Prediction Generation*, where, based on this neighborhood, predictions for items rated by the active user are produced.

IbCF is also based on the creation of neighborhoods. Yet, unlike the User-based filtering approach, those neighbors consist of similar items rather than similar users [8].

## 3 The HyCov Algorithm

In this section, we present a hybrid algorithm that keeps the core implementations of existing recommender systems and enhances them by adding a way to increase the percentage of items for which a filtering algorithm can generate predictions. In the following paragraphs we will describe how this general approach can be applied in the case of Item-based Collaborative Filtering, improving the coverage of their predictions, and, depending on the various parameter settings, leading to more accurate recommendations.

**The Item HyCov Implementation**

Let **R** be the $m \times n$ user-item ratings matrix, where element $r_{ij}$ denotes the rating that user $u_i$ (row $i$ from matrix **R**) gave to item $i_j$ (column $j$ from matrix **R**).

- *Step 1*: **Item Neighborhood Formation.** The basic idea in that step is to isolate couples of items, $i_j$ and $i_k$, which have been rated by a common user, and apply an appropriate metric to determine their similarity. We utilized the Adjusted Cosine Similarity approach, which, as shown in previous experiments [8], performs better than Cosine-based Similarity or Correlation-based Similarity.
  The formula for Adjusted Cosine Similarity of items $i_j$ and $i_k$ is the following:

$$sim_{jk} = adjcorr_{jk} = \frac{\sum_{i=1}^{m}(r_{ij} - \bar{r_i})(r_{ik} - \bar{r_i})}{\sqrt{\sum_{i=1}^{m}(r_{ij} - \bar{r_i})^2 \sum_{i=1}^{m}(r_{ik} - \bar{r_i})^2}} \quad (1)$$

  where $r_{ij}$ and $r_{ik}$ are the ratings that items $i_j$ and $i_k$ have received from user $u_i$, while $\bar{r_i}$ is the average of user's $u_i$ ratings. The summations over $i$ are calculated only for those of the $m$ users who have expressed their opinions over *both* items. Based on the calculated similarities, we form item neighborhood *IN*, which includes the $l$ items which share the greatest similarity with item $i_j$. Finally, we require that a possibly high correlation between the active item and a second random item is based on an adequate number of commonly rating users, known as *Common User Threshold*.
- *Step 2*: **User Neighborhood Formation.** User Neighborhood Formation is not part of the base algorithm of IbCF. It is implemented in the proposed approach for reasons that are explained in the following step of the procedure. The main purpose is to create a neighborhood of users most similar to the selected active user, $u_a$. We achieve that by simply applying *Pearson Correlation Similarity* as follows:

$$sim_{ai} = corr_{ai} = \frac{\sum_{j=1}^{n}(r_{aj} - \bar{r_a})(r_{ij} - \bar{r_i})}{\sqrt{\sum_{j=1}^{n}(r_{aj} - \bar{r_a})^2 \sum_{j=1}^{n}(r_{ij} - \bar{r_i})^2}} \quad (2)$$

where $r_{aj}$ and $r_{ij}$ are the ratings that item $i_j$ has received from users $u_a$ and $u_i$, while $\bar{r_a}$ and $\bar{r_i}$ are the average ratings of users' $u_a$ and $u_i$, respectively. The summations over $j$ are calculated only for those of the $n$ items which have been rated by *both* users. Now we can select the $p$ users who appear to have the greatest similarity to the active user, $u_a$, thus generating his neighborhood, $AN$. Again, we require that a possibly high correlation between the active user and a second random user is based on an adequate number of commonly rated items, known as *Common Item Threshold*.

- *Step 3*: **Prediction Generation.** The most crucial step of the recommendation procedure is Prediction Generation. At this step lies the main contribution of this hybrid approach.
  In the base algorithm of IbCF, a prediction of the active user's, $u_a$, rating on item $i_j$ is generated by computing the weighted sum of ratings given by $u_a$ on items belonging to the neighborhood of $i_j$ (Equation 3):

$$pr_{aj} = \frac{\sum_{k=1}^{l} sim_{jk} * r_{ak}}{\sum_{k=1}^{l} |sim_{jk}|}, \quad (3)$$

  where $sim_{jk}$ is the Adjusted Cosine Similarity between the active item, $i_j$, and an item, $i_k$, from its neighborhood, while $r_{ak}$ is the rating awarded by the active user to $i_k$.
  However, it is probable, especially when the dataset is sparse, that the active user hasn't rated *any* of the active item's neighbors. When that happens, the base algorithm is unable to generate a prediction for the item in mind.
  The idea behind the Item HyCov algorithm is that, instead of ignoring the specific item, and, consequently, accepting a reduction in the achieved Coverage, we can take into consideration what users similar to the active user, as expressed by belonging to his neighborhood, are thinking about item $i_j$. The proposed algorithm implements this idea by checking whether one or more neighbors of the active user, as calculated in the *User Neighborhood Formation* step, have expressed their opinion on the neighbor items. After identifying which user neighbors have rated the required items, the Item HyCov algorithm will utilize their ratings and generate a prediction for the active user, $u_a$, on the active item, $i_j$, by applying the following equation:

$$pr_{aj} = \frac{\sum_{k=1}^{l}\sum_{i=1}^{p} sim_{jk} * r_{ik}}{\sum_{k=1}^{l}\sum_{i=1}^{p} |sim_{jk}|} \quad (4)$$

  As shown in Equation 4, we generate a prediction for the active user $u_a$ by summing up the ratings of one or more of its $p$ neighbors ($\sum_{i=1}^{p}$) on the $l$ items as taken from the active item's $i_j$ neighborhood ($\sum_{k=1}^{l}$). The summations over $l$ are calculated only for those items which have been rated by at least one of the $p$

**Fig. 1** Item HyCov Pseudo-code (Prediction Generation)

```
1  PredictItemHyCov( u_a , i_j , AN, IN )
2  % u_a , i_j : active user/item % AN, IN: user/item neighborhood
3  NN ← ∅ %set of neighbor items rated by u_a or its neighbors
4  Foreach ( i_k , sim_jk ) ∈ IN
5     If ∃ r_ak Then next(NN) ← ( r_ak , sim_jk )
6  If NN ≠ ∅ Then Apply Equation (3)
7  Else
8     Foreach ( i_k , sim_jk ) ∈ IN
9        Foreach ( u_i , sim_ai ) ∈ AN
10          If ∃ r_ik Then next(NN) ← ( r_ik , sim_jk )
11    If NN ≠ ∅ Then Apply Equation (4)
```

neighbor users. Of course, users with zero correlation with the active user are excluded. The user ratings are weighted by the corresponding similarity, $sim_{jk}$, between the active item $i_j$ and the neighbor item $i_k$, with $k = 1, 2, ..., l$.

The pseudo-code of the prediction step of the Item HyCov is given in Figure 1.

- *Step 4*: **Measures of Performance.** Finally, two evaluation metrics are calculated, Mean Absolute Error and Coverage [9]. Mean Absolute Error (MAE) measures the deviation of predictions generated by the RS from the true rating values, as they were specified by the user. Coverage is computed as the fraction of items for which a prediction was generated over the total number of items that all available users have rated in the initial user-item matrix.

A similar approach could be followed in the case of User-based filtering. The main difference is that when the system cannot provide a prediction for the active item, its neighborhood is formulated through Item-based CF, and the ratings of neighbor users on these items are used for Prediction Generation.

## 4 Experimental Results

In this section we will evaluate the utility of the HyCov method. We will provide a brief description of the various experiments we executed and then we will present the results of these experiments.

For the execution of the subsequent experiments we utilized MovieLens, the dataset publicly available from the GroupLens research group [1]. The MovieLens dataset, used by several researchers [10, 2], consists of 100.000 ratings which were assigned by 943 users on 1682 movies. Ratings follow the 1(bad)-5(excellent) numerical scale. The sparsity of the data set is high, at a value of 93.7%. Starting from the initial data set, a distinct split of training (80%) and test (20%) data was generated.

At this point, it is necessary to note that while User-based and Item-based CF each had a couple of changing parameters (size of the user/item neighborhood and

common item/user threshold, correspondingly), the proposed hybrid approach has four free parameters, all of which can be altered during experiment execution: *user neighborhood size (p)* along with *common item threshold (cit)* in the stage of user neighborhood formation, and *item neighborhood size (l)* along with *common user threshold (cut)* in the stage of item neighborhood formation. Differences in MAEs and Coverages were compared using paired $t$-tests.

**Comparing the Item HyCov approach to Item-based Filtering**

The Item HyCov approach can be actually considered to be an enhancement of the plain Item-based filtering algorithm, since it proposes a way to increase the percentage of items for which the base algorithm can generate predictions. To support this claim, we include a couple of experiments that compare the performance of the two approaches.

For the first experiment, and specifically for the Item HyCov algorithm part, we kept the user neighborhood set to 18 users ($p = 18$), which, based on a series of preparatory experiments, displayed the best performing behavior for the combinations of the remaining parameters. The common item and common user thresholds were also fixed ($cit = 20$ and $cut = 10$). As for the base algorithm, *cut* was equal to 10. In both cases, the only changing parameter was the item neighborhood size.

Figure 2 depicts the results from this experiment. Figure 2(b) shows that Item HyCov actually improves on plain IbCF in terms of Coverage, for neighborhoods that include up to 100 items. This improvement is even more evident for small item neighborhoods where, because of the lack of sufficient neighbors, IbCF cannot generate an adequate number of predictions.

One might have expected that the inclusion of additional ratings would affect the prediction accuracy in a negative way. On the contrary, as one can see in Figure 2(a), the hybrid approach is constantly more accurate than plain IbCF, especially for neighborhoods including less than 50 items and for neighborhoods with more than 150 items.

A paired $t$-test indicated the statistical significance of differences ($p < 0.05$) at 95% confidence level between the base and the Item HyCov approaches (the normality requirement is met). The results suggest that the hybrid approach is significantly better than IbCF, both in terms of MAE ($t$ = 3.469, $p$ = 0.002) and Coverage ($t$ = -2.079, $p$ = 0.049).

For the second experiment, based on the MAE and Coverage results which were presented in the preceding section, we set the item neighborhood to include 50 items ($l = 50$) for both the hybrid and the plain IbCF approach. Specifically for the Item HyCov part of the experiment, the common item and common user thresholds were fixed ($cit = 20$ and $cut = 10$), the only changing parameter being the user neighborhood size. The basic idea behind this experiment was to test how different user neighborhood sizes would affect the overall system's performance. The results from this experiment are shown in Figure 3.

The advantage of Item HyCov in terms of Coverage is clear: it peaked at a value of 93.20% for neighborhoods including more than 8 users, whereas IbCF Coverage was equal to 91.31%. Regarding the observed MAE values, the best accuracy

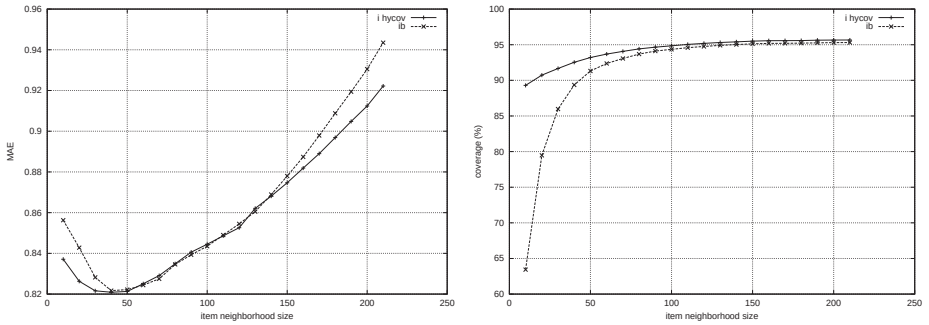


**Fig. 2** Comparing Item HyCov to IbCF for varying item neighborhood sizes in terms of (a)MAE and (b)Coverage

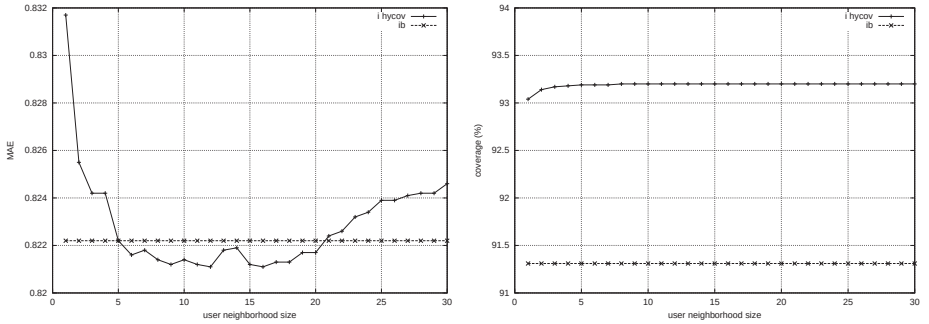


**Fig. 3** Comparing Item HyCov to IbCF for varying user neighborhood sizes in terms of (a)MAE and (b)Coverage

achieved by the hybrid approach was equal to 0.8211 for a user neighborhood of 12, while IbCF MAE for the same parameter settings was slightly worse, at 0.8222.

## 5 Conclusions

This paper presented a filtering algorithm which combines the strengths of two popular CF approaches, IbCF and UbCF, into a feature combination hybrid. Item HyCov attempts to deal with low prediction Coverage, a problem especially present in sparse datasets. The proposed approach was tested using the Movielens dataset and was compared with plain Item-based CF.

The experimental results indicated that the proposed approach significantly increases the prediction Coverage, with a simultaneous significant improvement of accuracy in terms of MAE. Another advantage of the present approach is that it requires no additional data and minimal additional implementation or modification of existing CF recommender systems. However, it must be noted that the application of

the hybrid approach may increase the computational cost of the prediction process up to $O(p^2 l)$, for $p$ neighbor users and $l$ neighbor items. This cost may be transferred to an off-line phase if item or user data change infrequently and therefore there is no practical need to perform these computations at prediction time.

Further research issues include the incorporation of dimensionality reduction methods as a preprocessing step, in order to deal with scalability problems (large number of users and/or items). Moreover, item or user demographic data could be utilized in the neighborhood formation stage [11]. Finally, further experiments could be carried out in order to investigate how datasets with different characteristics would affect the performance of the proposed algorithm.

## References

1. Grouplens, http://www.grouplens.org/ (accessed on october 2008).
2. H. J. Ahn. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences: an International Journal*, 178:37–51, 2008.
3. R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12:331–370, 2002.
4. J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22:5–53, 2004.
5. X. Jin, Y. Zhou, and B. Mobasher. A maximum entropy web recommendation system: Combining collaborative and content features. In *Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 612–617, Chicago, Illinois, 2005.
6. P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002)*, pages 187–192, Edmonton, Canada, 2001.
7. P. Resnick, N. Iacovou, M. Sushak, P. Bergstrom, and J. T. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, New York, NY, 1994.
8. B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl. Item-based collaborative filtering recommendation algorithms. In *10th International World Wide Web Conference (WWW10)*, pages 285–295, Hong Kong, 2001.
9. U. Shardanand and P. Maes. Social information filtering: Algorithms for automating 'word of mouth'. In *Proceedings of Computer Human Interaction*, pages 210–217, 1995.
10. S. Ujjin and P. J. Bentley. Particle swarm optimization recommender system. In *Proceedings of the IEEE Swarm Intelligence Sympoisum 2003*, pages 124–131, Indianapolis, 2003.
11. M. Vozalis and K. G. Margaritis. On the enhancement of collaborative filtering by demographic data. *Web Intelligence and Agent Systems, An International Journal*, 4(2):117–138, 2006.
12. J. Wang, A. P. deVries, and M. J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 501–508, Seattle, Washington, 2006.