Combining Content-based and Collaborative Filtering

Gabriela Polčicová and Pavol Návrat

Slovak University of Technology, Department of Computer Science and Engineering,
Bratislava, Slovakia,
{polcicova, navrat}@dcs.elf.stuba.sk

Abstract. In this paper we present a method for items recommendation. The proposed method uses estimated recommendations computed by Content-based filtering method in order to improve recommendations produced by Collaborative filtering method. By combining these two methods it is possible to alleviate the problem of lacking explicit ratings appearing in both of original methods.

Keywords

Information Filtering, Content-based Filtering, Collaborative Filtering.

1 Introduction

In recent years the amount of information from various research areas is growing in such way that users might become overflown by them. This has evoked increased activity in research area named Information Filtering (IF). Its aim is to help individuals or communities to find relevant information. This can be achieved by finding similar information to information of user's interest or by finding similar users who can recommend relevant information.

Our work focuses on IF methods that might be of interest for participants of EMGnet project. One aim of EMGnet project is to put together neurology experts as a community of individuals sharing a common area of interest. IF methods can help members of the community share knowledge about quality of information resources. IF system for patent clipping service in Human Media Research Laboratory in Japan [1] provides an example of applying such methods in a community of specialists.

We proposed a method that combines two filtering approaches, Content-based filtering (CBF) and Collaborative filtering. To evaluate and compare results of CF, CBF and a combined method we needed to have a dataset - set of items with parsable content and ratings from real users. As the dataset of patents and their ratings were not available we have used EachMovie dataset. The dataset consists of users ratings for movies and is often used for evaluating of filtering methods.

In the next section we present techniques for CBF, CF and combined approaches. In section 3 we describe data used in experiments. In section 4 are described experiments that allow comparissons of the methods. Sections 5 and 6 present results and discussion. The last section summarizes conclusions.

2 Content-based Filtering, Collaborative Filtering and their Combination

There are two main types of IF approaches, Content-based filtering (CBF) and Collaborative filtering (CF). Both methods use opinions of users on items in order to determine user's interests. Those opinions are mostly expressed by explicit ratings (numbers from a fixed scale). CBF and CF approaches use different data to represent users' interests (in users' *profiles*) and use different techniques for their judgments, but both try to estimate user's ratings for unrated information and based on those estimates form recommendations to the user.

In this section we describe approaches to CBF, CF and their combination, that we used in our work.

2.1 Content-based Approach

In CBF systems a user rates items according to their attractivity. The system analyzes content of those items and creates a profile that is a representation of user's interests in terms of items' content (keywords, phrases or features). Then CBF systems analyze content of items unknown to the user, compare it with the profile and estimate which of those items are interesting to the user (Figure 1). This method is effective for recommending textual documents, since it is able to recommend only those items, that are "understandable" to computers. System Lira [2] for recommending useful web-pages can be used as an example of a CBF system.

To represent items and profiles we use vector space. Each item i is represented by pair $\{(w_{i1}, \ldots, w_{in}), r_i\}$. Weights w_{ij} are weights for words from a given dictionary vector $\boldsymbol{D}$ for the item i and r_i is a user's rating for that item. To compute weights for words, item's words are stemmed and the most common and not informative words stored in *stop list* are erased. Then for each word j there is computed its weight w_{ij} using *term frequency - inverse document frequency*. To represent document maximum *max* words with highest weights were chosen and normalized. Profiles are made up from words of rated items. Used ratings are transformed so that the mean value is 0, the best rating is positive and the worst rating is negative. Then the weight of the word j in the profile is given by:

$$profile_j = \sum_{i=1}^{N} r'_i . w_{ij}, \tag{1}$$

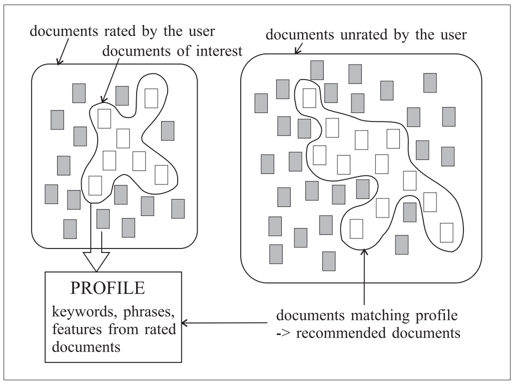

Fig. 1. Content-based filtering. User's profile is constituted from the content of documents rated by the user. Unrated documents matching profile are recommended to the user.

where N is the number of weighted items and r'_i is a transformed rating. The profile vector is normalized to length 1, too.

In order to measure how much a new item matches the profile we use cosine between angles of vectors representing item and profile, which is the most common measure used by CBF recommender systems. The value ranges from -1 to 1. We divide this interval into so many subintervals, as many values are in the rating scale. We assign each subinterval to one value from the scale, such that the lowest subinterval is assigned to the lowest value and the highest subinterval is assigned to the highest value. To each item there is assigned an estimate according to the subinterval, to which the weight of the value belongs. Items with value greater then or equal to threshold T_1 are recommended to the user. (See Appendix A)

2.2 Collaborative Approach

CF systems work by collecting users ratings on items. By comparing those ratings, they locate users who share common interests and make personalized recommendations based on like-minded users opinions. CF approach makes its judgments based on evaluations of items quality and it does not take formal content into account. The most known CF recommender systems are GroupLens system [6], [7] used for Usenent news domain and Ringo [9] for recommending music albums.

We used Correlation method (Appendix B), known as one of the most effective methods for collaborative filtering [4]. This method maintains a rating matrix $R = \{r_{ij}\}$, where r_{ij} is the rating of user i for item j. Rows of matrix R represent users' profiles. This matrix is used to find like-minded users for current user c by applying *Pearson correlation coefficient* k_{ci}

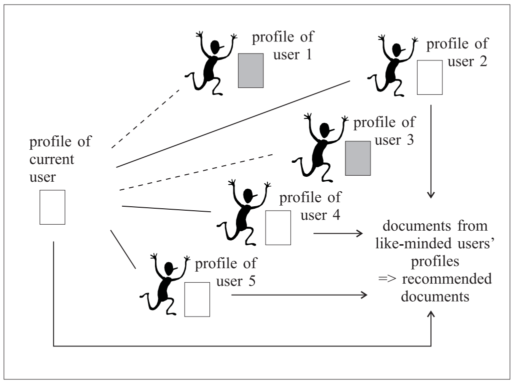

Fig. 2. Collaborative Filtering. From users in the community are chosen those who have similar preferences (similar profiles) and from their ratings are created recommendations to current user.

$$k_{ci} = \frac{\sum\limits_{j \in I_{ci}} (r_{cj} - \overline{r}_c)(r_{ij} - \overline{r}_i)}{\sqrt{\sum\limits_{j \in I_{ci}} (r_{cj} - \overline{r}_c)^2} \sqrt{\sum\limits_{j \in I_{ci}} (r_{ij} - \overline{r}_i)^2}}, \tag{2}$$

where I_{ci} is the set of items that were rated by both users c and i and $\overline{r}_i$ ($\overline{r}_c$ resp.) is the average rating of user i (c resp.). To compute estimates there are used not only ratings of users with similar interest (positive correlation), but also of users with opposite interests (negative correlation). Thus for new estimate computation for the current user we used ratings of users i such that $|k_{ci}| > T_2$. T_2 is a given threshold. To compute new estimates $\hat{r}_{cj}^{CF}$ is used weighted sum of ratings:

$$\hat{r}_{cj}^{CF} = \overline{r}_c + \frac{\sum\limits_{i \in U_{cj}} (r_{ij} - \overline{r}_i) k_{ci}}{\sum\limits_{i \in U_{cj}} |k_{ci}|}, \tag{3}$$

where U_{cj} is a set of users i who rated item j and $|k_{ci}| > T_2$.

Finally, items (j) with estimates satisfying $\hat{r}_{cj}^{CF} \geq T_1$ are recommended to the user c.

2.3 Combined Content-based and Collaborative Approach

Both CF and CBF methods have similar problems at the initial stage of their work - lack of ratings, that leads to inaccurate recommendations. This can discourage hopeful users. The greatest problem of CF method, known as the sparsity problem, is that there is available only a tiny fraction of all possible ratings, which leads to constriction of the number of those items the system is able to recommend. The problem of systems that use CBF method is precision of estimates and also recommendations. There is a hypothesis that combination of CF and CBF method can alleviate mentioned drawbacks of initial phase and make better recommendations. There were presented several approaches for combining CF and CBF methods ([3], [8], [5], [1]).

We propose a method for CF and CBF combination (Appendix C), where CBF estimates are used to fill up some missing ratings for CF method (Figure 3). To do this, both profiles - those based on item's content (CBF) and those based on ratings (CF) - are needed. Our method tries to alleviate the main CF drawback - the sparsity problem. This problem is the most limiting for the CF method in the initial phase, when there is still available only a small number of ratings. In order to measure similarity between two users we use two kinds of coefficients. The first one is Pearson correlation coefficient k_{ci} (eq. 2), that is computed from ratings of users c and i. The second one is Pearson correlation coefficient k'_{ci} that is computed from both ratings and some CBF estimates. CBF estimates are estimates computed by the pure CBF method. To compute k'_{ci} we take into account only those items that were rated by at least one of the comparing users. Thus we consider only these situations:

1. both users rated an item,
2. only one of two users rated an item and CBF estimate is taken instead of the other one's rating.

Items that were not rated by any of the two users are not used in k'_{ci} computation. Estimates are computed by equation derived from (3):

$$\hat{r}_{cj} = \overline{r}_c + \frac{\sum\limits_{i \in U_{cj}} (r_{ij} - \overline{r}_i)\, k_{ci} + \sum\limits_{i \in U'_{cj}} \left(\hat{r}_{ij}^{CBF} - \overline{r}_i\right) k'_{ci}}{\sum\limits_{i \in U_{cj}} |k_{ci}| + \sum\limits_{i \in U'_{cj}} |k'_{ci}|}. \tag{4}$$

U'_{cj} is the set of users i who rated and item j and $|k'_{ci}| > T_2$. $\hat{r}_{ij}^{CBF}$ is either rating for item j from user i, if available, or estimate computed by CBF method. There are used only ratings and estimates of those users whose coefficients (k_{ci} or k'_{ci}) are greater then T_2 in all the sums of the eq. 4.

3 Dataset

For experiments we use EachMovie database[1] and Internet Movie Database[2] (IMDB). EachMovie contains explicit ratings for movies. This was developed by Digital Equipment Research Center in years 1995 - 1997. For CBF we use movies

[1] http://www.research.digital.com/SRC/eachmovie/

[2] http://www.imdb.com/

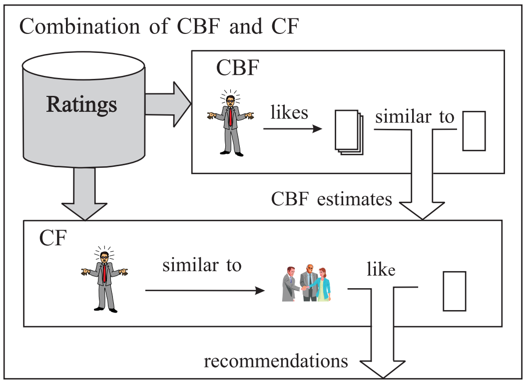

Fig. 3. Schema of Combined CF and CBF method. In order to make recommendations by CF method, both ratings and CBF estimates are used.

description from IMDB developed by Amazon.com company.

Since we focus on performance of the methods in the initial phase under conditions where only a small number of ratings is available, we provide our analysis on three smaller datasets A, B and C created from the EachMovie data. Those datasets consist of all ratings collected in three periods. The description of the datasets is in Table 1. Character of the datasets (including ratings from users that go to the cinema to see the movies they suppose to be interesting for them) imply their distribution of ratings (Figure 4). To see the overall performance of tested method we tested also a simple *Constant method* which for each rating in a dataset assigns estimate equal to number 5 that is the most often occurring rating in a dataset.

Table 1. Description of the datasets used in experiments.

dataset	period	number of ratings	number of users
A	Jan 9, 1995 - Mar 1, 1996	810	71
B	Jan 9, 1995 - Mar 15, 1996	2407	131
C	Jan 9, 1995 - Apr 1, 1996	12290	651

4 Experiments

4.1 Description of the Experiments

We made several preliminary experiments with datasets A, B and C (Table 1) divided to training and test sets. For each dataset we made ten experiments with

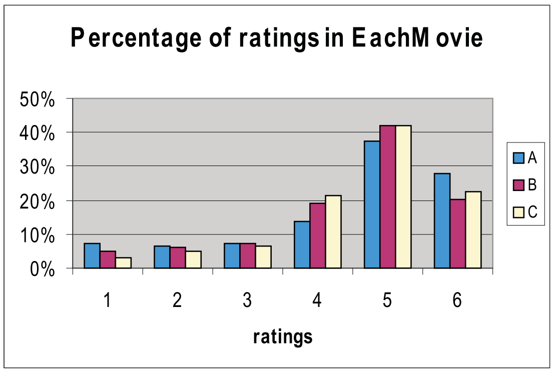

Fig. 4. Distribution of ratings in EachMovie dataset.

different test and training sets. And in all such experiments we tested CF, CBF, the combined and the constant method (test and training sets were identical for all methods in each experiment). For each dataset and for each metric, except Precision and Recall, we analyzed results using 1-way ANOVA technique with 95% confidence level.

4.2 Parameter Settings

In experiments we take 90% of used data as a training set and 10% as a test set. Test set is built up randomly so that each user, whose rating is chosen, has at least 5 ratings in the training set. This was set up in order to ensure that methods try to make recommendations only for those users, whose interests they know. In order to reduce memory requirement and avoid over-fitting we set up the maximal number of words, representing the document, *max* to 100 in the CBF estimates computation. We set threshold $T_1 = 4$, thus only items with estimates greater then the mean value of the scale (4, 5 and 6) are recommended to the user. Threshold T_2 is set to 0.5, which means that only ratings of users i having $|k_{ci}| > 0.5$ are used for computation of user c's estimates. This was chosen similarly as in the Ringo system [9], where it was the mean value of the tested thresholds. To create estimates by CBF method we uniformly divide an interval $< -1, 1 >$ to 6 subintervals since we use the 6-point scale. According to the subinterval, to which the similarity weight belongs to, estimate is assigned.

5 Results

5.1 Metrics

We use six metrics in order to measure quality of estimates, since each one measures the quality of the methods from different point of view. In following equations, S_1 is a set of recommended items and S_2 is a set of liked items.

- *Coverage* is a percentage of items for which the method is able to compute estimates,
- *Normalized Mean absolute error* (NMAE), is a measure of the normalized deviation of estimates from their true user-specified ratings given by

$$\text{NMAE} = \frac{\sum\limits_{(i,j)\in T} |r_{ij} - p_{ij}|}{n\ s}, \tag{5}$$

  where T is a test set, n is the number of items in T and s is the number of points in the scale,
- *Accuracy* is a percentage of items that are correctly classified:

$$\text{Accuracy} = \frac{|S_1 \cap S_2| + |\overline{S_1} \cap \overline{S_2}|}{|S_2| + |\overline{S_2}|}, \tag{6}$$

- *Precision* is a percentage of recommended items that are relevant:

$$\text{Precision} = \frac{|S_1 \cap S_2|}{|S_1|}, \tag{7}$$

- *Recall* is a percentage of relevant items that are recommended:

$$\text{Recall} = \frac{|S_1 \cap S_2|}{|S_2|}, \tag{8}$$

- *F-measure* is a weighted combination of Precision and Recall and is often used because it is easy to optimize either of them separately. That is why we do not analyze resutls measured by Precision and Recall and we analyze only F-measure. It is given by:

$$\text{F-measure} = \frac{2.\text{Precision}.\text{Recall}}{\text{Precision} + \text{Recall}}. \tag{9}$$

5.2 Preliminary Results

The graph on Figure 5 presents results measured by *Coverage* for three datasets and for each dataset for four methods (CF, CBF, combined and constant). The outcomes for all datasets show significant improvement of CBF, combined and constant methods as compared to CF method. Table 2 shows differences of means for results that are significant. This means that also combined method can alleviate the sparsity problem, what we hypothesized.

Graph on Figure 6 shows that CF, combined and constant methods provide significant increase in *Normalized mean absolute error* as compared to CBF method. Significant differences between means are presented in Table 3.

Accuracy provides no significant improvements of any from the methods (Figure 7). Since we analyzed results of *F-measure* (Figure 8) we did not analyze results of *Precision* and *Recall*. Only the constant method provides significant improvement of results for F-measure as compared to combined method only for A dataset (means difference is 0.03622).

Table 2. Coverage: differences between means for methods that are significant.

methods	dataset A	dataset B	dataset C
CBF vs CF	0.137	0.141	0.142
combined vs CF	0.163	0.165	0.165
constant vs CF	0.035	0.037	0.037

Table 3. Normalized Absolute Mean Error: differences between means for methods that are significant.

methods	dataset A	dataset B	dataset C
CBF vs CF	0.02841	0.03892	0.04860
CBF vs combined	0.03295	0.04033	0.04760
CBF vs constant	0.05046	0.03317	0.04331
CF vs constant	0.02205	-	-0.00529
combined vs constant	0.01751	-	-

6 Discussion

The results show that combined method improves coverage of CF method and that provides smaller mean absolute error than CBF method. It is interesting that simple Constant method provides comparable results. From this we deduce that since the results show that the method combining CF and CBF can be helpful in the initial phase, they also indicate that the concept of testing might be unsuitable for such problems.

7 Conclusions

We proposed a method that combines CF and CBF methods for information recommendation. We made several preliminary experiments by using EachMovie dataset in order to compare results of CF, CBF, combined and simple constant method which predicts the most often occurring rating in the dataset. The results showed that combined method can improve recommendations (coverage of CF method and mean absolute error of CBF method) in the phase when only a small number of ratings is available. But the constant method provides surprisingly convenient results what might indicate that the concept of testing is not appropriate for such problems.

8 Acknowledgements

This work was supported by the Slovak Science Grant Agency grant 1/7611/20 and Copernicus CP PL 977069S.

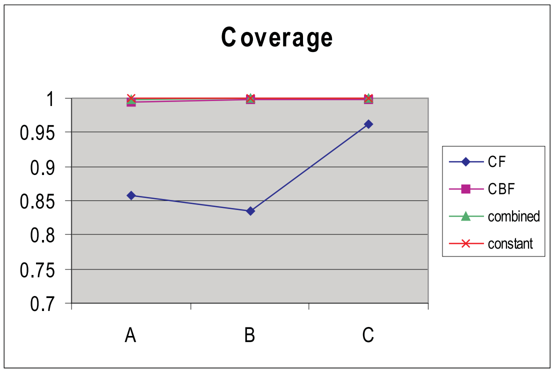

Fig. 5. Coverage for experiments with three datasets (A, B and C) and for each of them for four methods (CF, CBF, combined and constant)h.

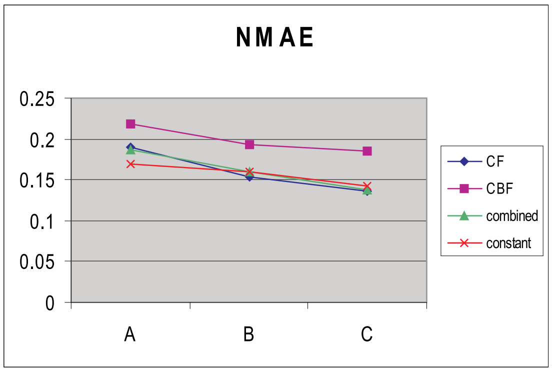

Fig. 6. Normalized Mean Absolute Error for experiments with three datasets (A, B and C) and for each of them for four methods (CF, CBF, combined and constant)i.

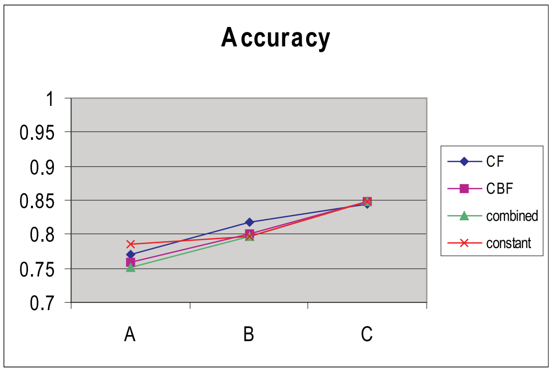

Fig. 7. Accuracy for experiments with three different datasets (A, B, and C) and for each of them for four methods (CF, CBF, combined and constant)j.

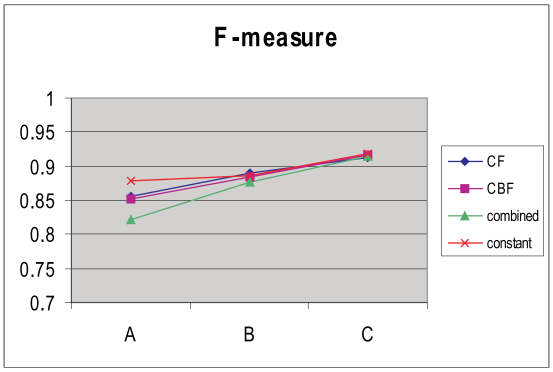

Fig. 8. F-measure for experiments with three different datasets (A, B, and C) and for each of them for four methods (CF, CBF, combined and constant)k.

References

1. Y. Ariyoshi Improvement of Combination Information Filtering Method Based on Reliabilities. In *Machine Learning for Information Filtering at IJCAI 99*, 1999.
2. M. Balabanovic and Y. Shoham. Learning Information Retrieval Agents: Experiments with automated web browsing. In *AAAI 1995 Spring Symposium of Information Gathering from Heterogeneous, Distributed Environments, Stanford*, 1995. url: *http://robotics.stanford.edu/people/marko/papers/lira.ps.*
3. M. Balabanovic, and Y. Shoham. Fab: Content-based, Collaborative Recommendation. *Communications of the ACM*, 40(3):29–38, 1997.
4. J.S. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the Fourteenth Conference on Uncertainity in Artificial Intelligence*, Madison, Morgan Kaufmann Publisher, 1998.
5. N. Good, J.B. Schafer, J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, Combining Collaborative Filtering with Personal Agents for Better Recommendations. In *Proceedings of the 1999 Conference of the American Association of Artifical Intelligence (AAAI-99)*. pp 439-446, 1999.
6. J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon and J. Riedl: GroupLens - Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, 40(3):77–87, 1997.
7. P. Resnick, N. Iacovou, M. Sushak, P. Bergstrom and J. Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 Computer Supported Cooperative Work Conference*, New York, 1994. ACM.
8. B. Sarwar, J. Konstan, A. Borchers, J. Herlocker, B Miller, and J. Riedl. Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. In *Proceedings of the 1998 Conference on Computer Supported Cooperative Work Nov.* 1998.
9. U. Shardanand and P. Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of the 1995 ACM Conference on Human Factors in Computing Systems*, str. 210–217, New York, 1995. ACM.

Appendix

A CBF method

```
CBF()
begin
    /* Initialization */
    for each item i <- 1 to NUMBER_OF_ITEMS do begin
        for each word w <- 1 to NUMBER_OF_KEYWORDS do
            Items[i][w] <- weight of a word w in item i
        normalize(Item[i])
    endfor
    for each user c <- 1 to NUMBER_OF_USERS do begin
        for each item i <- 1 to NUMBER_OF_ITEMS do begin
            if user c rated item i and rating is not in the
                Test set then
                Rating[i] <- rating of item i from user c
            else Rating[i] <- 0
            if rating of user c for item i is in Test set then
                Test_set[i] <- 1
            else Test_set[i] <- 0
        endfor
        test <- 0
        for i <- 1 to NUMBER_OF_ITEMS do
            test <- test + Test_set[i]
        if test > 0 then begin
            /* Profile forming */
            for each item i <- 1 to NUMBER_OF_ITEMS do
                if Rating[i] <> 0 then
                    for each word w <- 1 to NUMBER_OF_KEYWORDS do
                        Profile[w] <- Rating[i] * (weight[w] - scale_mean)
            normalize(Profile)
            /* Estimate computation */
            for each item i <- 1 to NUMBER_OF_ITEMS do
                if Test_set[i] = 1 then begin
                    for each word w <- 1 to NUMBER_OF_KEYWORDS do
                        s <- s + Profile[w] . Items[i][w]
                    CBF_estimate[c][i] <- assign_estimate(s)
                endif
        endif
    endfor
end
```

B CF method

```
CF(Threashold T_2)
```

```
begin
    /* Initialization */
    for each user u <- 1 to NUMBER_OF_USERS do begin
        for each item i <- 1 to NUMBER_OF_ITEMS do
            if user u rated item i and rating is not in the
                Test set then
                Ratings[u][i] <- rating of item i from user u
            else Ratings[u][i] <- 0
        avg[u] <- average_of_ratings_of_user(Ratings[u])
    endfor
    /* Estimates computation */
    for each user c <- 1 to NUMBER_OF_USERS do begin
        for each item i <- 1 to NUMBER_OF_ITEMS do begin
            if rating of user c for item i is in Test set then
                Test_set[i] <- 1
            else Test_set[i] <- 0
        test <- 0
        for each item i <- 1 to NUMBER_OF_ITEMS do
            test <- test + Test_set[i]
        if test > 0 begin
            /* Profile forming */
            for each user u <- 1 to NUMBER_OF_USERS do
                k[u] <- Pearson_correlation_coefficient(Ratings[c],
                        Ratings[u])
             for each item i <- 1 to NUMBER_OF_ITEMS do
               if Test_set[i] = 1 then begin
                   sum1 <- 0
                   sum2 <- 0
                   for each user u <- 1 to NUMBER_OF_USERS do
                       if Ratings[u][i] > 0 and k[u] > T_2 then begin
                           sum1 <- sum1 +
                                   (Ratings[u][i] - avg[u]).k[u]
                           sum2 <- sum2 + |k[u]|
                       endif
                   CF_estimate[c][i] <- avg[c] + sum1 / sum2
               endif
        endif
    endfor
end
```

C combined method

```
CF_CBF_combination(Threashold T_2)
begin
    /* Initialization */
    for each user u <- 1 to NUMBER_OF_USERS do
```

```
    for each item i <- 1 to NUMBER_OF_ITEMS do begin
        if user u rated item i and rating is not in the
            Test set then
            Ratings[u][i] <- rating of item i from user u
        else Ratings[u][i] <- 0
        Estimates[u][i] <- Ratings[u][i]
    endfor
/* CBF part */
for each user u <- 1 to NUMBER_OF_USERS do
    for each item i <- 1 to NUMBER_OF_ITEMS do
        if Ratings[u][i] = 0 then
            Estimates[u][i] <- rating computed by CBF()
/* CF part: Estimates computation */
for each user c <- 1 to NUMBER_OF_USERS do begin
    for each item i <- 1 to NUMBER_OF_ITEMS do begin
        if rating of user c for item i is in Test set then
            Test_set[i] <- 1
        else Test_set[i] <- 0
    test <- 0
    for i <- 1 to NUMBER_OF_ITEMS do
        test <- test + Test_set[i]
    if test > 0 begin
        /* Profile forming */
        for each user u <- 1 to NUMBER_OF_USERS do
            k_r[u] <- Pearson_correlation_coefficient(Ratings[c],
                           Ratings[u])
            k_e[u] <- Pearson_correlation_coefficient(Estimates[c],
                           Estimates[u])
         for each item i <- 1 to NUMBER_OF_ITEMS do
            if Test_set[i] = 1 then begin
                sum1 <- 0
                sum2 <- 0
                for each user u <- 1 to NUMBER_OF_USERS do begin
                    if Ratings[u][i] > 0 and k_r[u] > T_2 then begin
                        sum1 <- sum1 +
                                (Ratings[u][i] - avg[u]).k_r[u]
                        sum2 <- sum2 + |k_r[u]|
                    endif
                    if Estimates[u][i] and k_p[u] > T_2 then begin
                        sum1 <- sum1 +
                                (Ratings[u][i] - avg[u]).k_p[u]
                        sum2 <- sum2 + |k_p[u]|
                    endif
                endfor
                CF_estimate[c][i] <- avg[c] + sum1 / sum2
```

```
            endif
        endif
    endfor
end
```