

# On a Probabilistic Combination of Prediction Sources

Ioannis Rousidis, George Tzagkarakis, Dimitris Plexousakis, and Yannis Tzitzikas

Institute of Computer Science, FORTH, Heraklion, Greece  
{rousidis,gtzag,dp,tzitzik}@ics.forth.gr

**Abstract.** Recommender Systems (RS) are applications that provide personalized advice to users about products or services they might be interested in. To improve recommendation quality, many hybridization techniques have been proposed. Among all hybrids, the weighted recommenders have the main benefit that all of the system's constituents operate independently and stand in a straightforward way over the recommendation process. However, the hybrids proposed so far consist of a linear combination of the final scores resulting from all recommendation techniques available. Thus, they fail to provide explanations of predictions or further insights into the data. In this work, we propose a theoretical framework to combine information using the two basic probabilistic schemes: the sum and product rule. Extensive experiments have shown that our purely probabilistic schemes provide better quality recommendations compared to other methods that combine numerical scores derived from each prediction method individually.

**Keywords:** Recommender Systems, Collaborative Filtering, Personalization, Data Mining.

## 1 Introduction

Nowadays, most of the popular commercial systems use collaborative filtering (CF) techniques to efficiently provide recommendations to users based on opinions of other users [7][9][10]. To effectively formulate recommendations, these systems rely either upon statistics (user ratings) or upon contextual information about items.

CF, which relies on statistics, has the benefit of learning from information provided by a user and other users as well. However, RS could suffer from the sparsity problem: accurate recommendations cannot be provided unless enough information has been gathered. Other problems and challenges that RS have to tackle include: (a) the user bias from rating history (statistics), (b) the "gray sheep" problem, where a user cannot match with anyone of the other users' cliques, and (c) the cold-start problem, where a new item cannot be recommended due to the lack of any information. These problems reduce the strength of statistics-based methods. On the other hand, if RS rely merely on the content of the items, then they tend to recommend only items with content similar to those already rated by a user.

The above observations indicate that in order to provide high quality recommendations we need to combine all sources of information that may be available. To this

end, we propose a purely probabilistic framework introducing the concept of uncertainty with respect to the accurate knowledge of the model.

Recently, hybridization of recommendation techniques has been an interesting topic, since recommenders have different strengths over the space. In [4], two main directions on combining recommenders are presented: the first combines them in a row by giving different priorities to each one and passing the results of one as an input to the other while the second applies all techniques equally and finds a heuristic to produce the output. Each hybrid has its tradeoff. According to [4], the latter hybrids, especially the weighted recommenders, have the main benefit that all of the system's capabilities are brought to bear on the recommendation process in a straightforward way and it is easy to perform post-hoc credit assignment and adjust the hybrid accordingly. Previous works in this area [2][5][11][15][16][17] performed merging of recommendation sources as a naïve linear combination of the numerical results provided by each recommendation technique individually. In general, these approaches are not capable of providing explanations of predictions or further insights into the data. Our approach differs from the others in that the combination of distinct information sources has a pure and meaningful probabilistic interpretation, which may be leveraged to explain, justify and augment the results.

The paper is organized as follows: in Section 2, we present background theory on the predictions techniques to be used. In Section 3, we introduce the two basic probabilistic schemes for combining information sources. In Section 4, we evaluate the models resulting from our framework to conclude our work in Section 5.

## 2 Prediction Techniques

Many approaches for CF have been previously proposed, each of which treats the problem from a different angle, and particularly by measuring similarity between users [3][7][13] or similarity between items [6][14]. Heuristics, such as *k-nearest neighbors* (KNN), have been used when the existence of common ratings between users is required in order to calculate similarity measures. Thus, users with no common items will be excluded from the prediction procedure. This could result in a serious degradation of the coverage of the recommendation, that is, the number of items for which the system is able to generate personalized recommendations could decrease.

In a recent work [12], a hybrid method combining the strengths of both model-based and memory-based techniques outperformed any other pure memory-based as well as model-based approach. The so-called *Personality Diagnosis* (PD) method is based on a simple probabilistic model of how people rate titles. Like other model-based approaches, its assumptions are explicit, and its results have a meaningful probabilistic interpretation. Like other memory-based approaches it is fairly straightforward, operating over all data, while no compilation step is required for new data. The following section contains a description of the PD algorithm; moreover we provide an extension of this approach to an item-based and a content-based direction.

### 2.1 Personality Diagnosis

PD states that each user  $u_i$ , where  $i=1,2,\dots,m$ , given any rating information over the objects available, has a *personality type* which can be described as.

$$P_{u_i}^{true} = \{r_{i,1}^{true}, r_{i,2}^{true}, \dots, r_{i,n}^{true}\} \quad (1)$$

where  $P_{u_i}^{true}$  is user's  $u_i$  vector of "true" ratings  $r_{i,j}^{true}$  over observed objects  $o_j$ . These ratings encode users' underlying, internal preferences. Besides, we assume the existence of a critical distinction between the true and reported ratings. In particular, the true ratings  $P_{u_i}^{true}$  cannot be accessed directly by the system, while the reported ratings, which are provided to the system, constitute the only accessible information. In our work, we consider that these ratings include Gaussian noise based on the fact that the same user may report different ratings depending on different occasions, such as the mood, the context of other ratings provided in the same session or on any other reason - external factor. All these factors are summarized as a Gaussian noise. Working in a statistical framework, it is assumed that a user's  $u_i$  actual rating  $r_{i,j}$  over an object  $o_j$ , is drawn from an independent normal distribution with mean  $r_{i,j}^{true}$ , which represents the true rating of the  $i^{\text{th}}$  user for the  $j^{\text{th}}$  object. Specifically:

$$\Pr(r_{i,j} = x | r_{i,j}^{true} = y) \propto e^{-(x-y)^2/2\sigma^2} \quad (2)$$

Where  $x, y \in \{1, \dots, |r|\}$  and  $|r|$  denotes the number of possible rating values. It is further assumed that the distribution of rating vectors (personality types), which are contained in the rating matrix of the database, is representative of the distribution of personalities in the target population of users. Based on this, the prior probability  $\Pr(P_{u_a}^{true} = \kappa)$  that the active user  $u_a$  rates items according to a vector  $\kappa$ , is given by the frequency that other users rate according to  $\kappa$ . So, instead of counting occurrences explicitly, a random variable  $P_{u_a}^{true}$  is defined which takes one out of  $m$  possible values,  $P_{u_1}^{true}, P_{u_2}^{true}, \dots, P_{u_m}^{true}$  each one with equal probability  $1/m$ . Thus, given the ratings of a user  $u_a$ , we can apply Bayes' rule to calculate the probability that he is of the same personality type as any other user  $u_i$ , with  $i \neq a$ :

$$\begin{aligned} & \Pr(P_{u_a}^{true} = P_{u_i}^{true} | r_{a,1} = x_1, \dots, r_{a,n} = x_n) \\ & \propto \Pr(r_{a,1} = x_1 | r_{a,1}^{true} = r_{i,1}) \dots \Pr(r_{a,n} = x_n | r_{a,n}^{true} = r_{i,n}) \cdot \Pr(P_{u_a}^{true} = P_{u_i}^{true}) \end{aligned} \quad (3)$$

Once we have computed this quantity for each user  $u_i$ , we can find the probability distribution of user's  $u_a$  rating for an unobserved object  $o_j$ , as follows:

$$\begin{aligned} & \Pr(r_{a,j} = x_j | r_{a,1} = x_1, \dots, r_{a,n} = x_n) \\ & \propto \sum_{i=1}^m \Pr(r_{a,j} = x_j | r_{a,j}^{true} = r_{i,j}) \cdot \Pr(P_{u_a}^{true} = P_{u_i}^{true} | r_{a,1} = x_1, \dots, r_{a,n} = x_n) \end{aligned} \quad (4)$$

where  $r_{a,j} \in \{1, \dots, |r|\}$ . The algorithm has a time and space complexity of the order  $O(mn)$ , as do the memory-based methods. According to the PD method, the observed ratings can be thought of as "symptoms", while each personality type, whose probability to be the cause we examine, as a "disease".

## 2.2 Feature Diagnosis

If we rotate the rating matrix by  $90^\circ$  we may consider the problem of recommendation formulation from another point of view introducing the notion of Feature Diagnosis (FD). Based on that, for any object  $o_i$ , where  $i=1,2,\dots,n$ , and given any rating information from users available, a type of features can be described as:

$$F_{o_i}^{true} = \{r_{1,i}^{true}, r_{2,i}^{true}, \dots, r_{m,i}^{true}\} \quad (5)$$

where  $F_{o_i}^{true}$  is object's  $o_i$  vector of “true” ratings  $r_{j,i}^{true}$  derived from users  $u_j$ . Thus, here we assume that these ratings include Gaussian noise based on the fact that ratings of the same user on different items may be temporally related (i.e., if their popularities behave similarly over time). For example, during the period near St. Valentines' day, romance movies may be more popular than movies about war. All these factors are summarized as a Gaussian noise. These ratings encode object's underlying, internal type of features. As in PD, it is again assumed that the distribution of rating vectors (feature types) is representative of the distribution of features in the target population of objects. So, instead of counting occurrences explicitly, a random variable  $F_{o_a}^{true}$  is defined that takes one out of  $n$  possible values,  $F_{o_1}^{true}, F_{o_2}^{true}, \dots, F_{o_n}^{true}$ , each one with equal probability  $1/n$ . Finally, given the ratings of an object  $o_a$ , we can apply Bayes' rule to calculate the probability to be of the same feature type as any object  $o_i$ , with  $i \neq a$ :

$$\Pr(F_{o_a}^{true} = F_{o_i}^{true} | r_{1,a} = x_1, \dots, r_{m,a} = x_m) \\ \propto \Pr(r_{1,a} = x_1 | r_{1,i}^{true} = r_{1,i}) \dots \Pr(r_{m,a} = x_m | r_{m,i}^{true} = r_{m,i}) \cdot \Pr(F_{o_a}^{true} = F_{o_i}^{true}) \quad (6)$$

Once we have computed this quantity for each object  $o_i$ , we can find the probability distribution for user's  $u_j$  rating of an unobserved object  $o_a$  using the following expression:

$$\Pr(r_{j,a} = x_j | r_{1,a} = x_1, \dots, r_{m,a} = x_m) \\ \propto \sum_{i=1}^n \Pr(r_{j,a} = x_j | r_{j,i}^{true} = r_{j,i}) \cdot \Pr(F_{o_a}^{true} = F_{o_i}^{true} | r_{1,a} = x_1, \dots, r_{m,a} = x_m) \quad (7)$$

According to FD, the observed ratings can be thought of as “*symptoms*”, while the features type as “*populations*” where symptoms may develop. The algorithm has a time and space complexity of the order  $O(mn)$ .

### 2.3 Context Diagnosis

The context of the objects, e.g. for a movie recommender any textual information on genres, can also provide useful information for recommendations. For this purpose, we define the following context vector.

$$C_{o_i}^{true} = \{c_{i,1}^{true}, c_{i,2}^{true}, \dots, c_{i,k}^{true}\} \quad (8)$$

where  $C_{o_i}^{true}$  is the “true” context type of the object  $o_i$  according to  $k$  categories. We assume that the probability of two objects to be of the same context type, taking into account the categories in which they belong, can be derived by associating their context vectors. We calculate this probability with the following expression:

$$\Pr(C_{o_a}^{true} = C_{o_i}^{true} | c_{a,1}, c_{a,2}, \dots, c_{a,k}) \propto \frac{|C_{o_a} \cap C_{o_i}|}{\max(|C_{o_a}|, |C_{o_i}|)} \cdot \Pr(C_{o_a}^{true} = C_{o_i}^{true}) \quad (9)$$

where  $c_{o,i}$  defines the membership of object  $o$  to category  $i$  (e.g. a 0 or 1 in an item-category bitmap matrix). The distribution of the category vectors (context types) of the objects which is available in the category matrix of the database is assumed to be representative of the distribution of context types in the target population of objects. Assuming again equal probability  $1/n$ , we can find the probability distribution for user's  $u_j$  rating of an unobserved object  $o_a$  based upon its context type is as:

$$\Pr(r_{j,a} = x_j | c_{a,1}, c_{a,2}, \dots, c_{a,k}) \propto \sum_{i=1}^n \Pr(r_{j,a} = x_j | r_{j,a}^{true} = r_{j,i}) \cdot \Pr(C_{o_a}^{true} = C_{o_i}^{true} | c_{a,1}, c_{a,2}, \dots, c_{a,k}) \quad (10)$$

The algorithm has a time and space complexity of the order  $O(n)$ , considering the number  $k$  of all categories available as a constant. According to CD, the observed ratings can be thought of as “*symptoms*” which may be developed in certain “*categories*” defined by a gamut of contextual attributes.

### 3 Combination Strategies

In probability theory, we can find two basic combinatorial schemes for the combination of distinct information sources, namely, the product-rule and the sum-rule. In this section, we show how this theory can be applied in our case where we aim to combine the prediction techniques presented in Section 2. The described framework is purely probabilistic and we argue this is the major advantage compared to the previous works. The traditional combination used widely so far is also presented.

#### 3.1 Product Rule

According to the product rule, we assume that the three types of information used to make predictions are independent. We apply Bayes’ rule assuming that the probability of a rating value to be the predicted value is conditioned on the ratings of the user, the ratings of the object and the categories that the object belongs to, thus:

$$\Pr(r_{i,j} | P_{u_i}^{true}, F_{o_j}^{true}, C_{o_j}^{true}) = \frac{\Pr(P_{u_i}^{true}, F_{o_j}^{true}, C_{o_j}^{true} | r_{i,j}) \Pr(r_{i,j})}{\Pr(P_{u_i}^{true}, F_{o_j}^{true}, C_{o_j}^{true})} \quad (11)$$

On the equation above we neglect the denominator, which is the unconditional measurement of the joint probability density, since it is common to all rating values  $r_{i,j}$  which we also consider to have equal probability. Thereby we focus only on the first term of the numerator which represents the conditional joint probability measurement distribution extracted by all the “true” vectors. We initially assumed that these vectors are conditionally independent, so:

$$\Pr(P_{u_i}^{true}, F_{o_j}^{true}, C_{o_j}^{true} | r_{i,j}) = \Pr(P_{u_i}^{true} | r_{i,j}) \Pr(F_{o_j}^{true} | r_{i,j}) \Pr(C_{o_j}^{true} | r_{i,j}) \quad (12)$$

Finally, by applying Bayes’ rule to each one of the factors of Eq(12) we obtain the probability of a rating value as:

$$\Pr(r_{i,j} | P_{u_i}^{true}, F_{o_j}^{true}, C_{o_j}^{true}) \propto \Pr(r_{i,j} | P_{u_i}^{true}) \Pr(r_{i,j} | F_{o_j}^{true}) \Pr(r_{i,j} | C_{o_j}^{true}) \quad (13)$$

The argument that maximizes this expression indicates the rating that user  $u_i$  is most likely to assign to object  $o_j$ .

#### 3.2 Sum Rule

In order to combine both PD and FD we introduce a binary variable  $B$  that refers to the relative influence of each method. When  $B$  is equal to 1, the prediction comes only from user’s rating vector, while when  $B$  is equal to 0 indicates full dependency on the

object's rating vector. Under these assumptions, the conditional probability can be computed by marginalization on the binary variable  $B$ . Therefore, the probability distribution of objects  $o_j$  rating by user  $u_i$  is given by

$$\begin{aligned} \Pr(r_{i,j} | P_{u_i}^{true}, F_{o_j}^{true}) &= \sum_B \Pr(r_{i,j} | P_{u_i}^{true}, F_{o_j}^{true}, B) \Pr(B | P_{u_i}^{true}, F_{o_j}^{true}) \\ &= \Pr(r_{i,j} | P_{u_i}^{true}, F_{o_j}^{true}, B=1) \Pr(B=1 | P_{u_i}^{true}, F_{o_j}^{true}) \\ &\quad + \Pr(r_{i,j} | P_{u_i}^{true}, F_{o_j}^{true}, B=0) \Pr(B=0 | P_{u_i}^{true}, F_{o_j}^{true}) \end{aligned} \quad (14)$$

By definition,  $r_{i,j}$  is independent from user's ratings when  $B=0$  so  $\Pr(r_{i,j} | P_{u_i}^{true}, F_{o_j}^{true}, B=0) = \Pr(r_{i,j} | F_{o_j}^{true})$ . The opposite holds when  $B=1$ , that is,  $\Pr(r_{i,j} | P_{u_i}^{true}, F_{o_j}^{true}, B=1) = \Pr(r_{i,j} | P_{u_i}^{true})$ . If we use a parameter  $\vartheta$  to denote the probability  $\Pr(B=1 | P_{u_i}^{true}, F_{o_j}^{true})$  we have:

$$\Pr(r_{i,j} | P_{u_i}^{true}, F_{o_j}^{true}) = \Pr(r_{i,j} | P_{u_i}^{true}) \vartheta + \Pr(r_{i,j} | F_{o_j}^{true}) (1 - \vartheta) \quad (15)$$

To include any contextual information about the object in our conditional hypothesis we introduce another binary variable which takes the values 0 when the prediction depends solely on ratings and 1 when it relies only on the context. So, by marginalizing the binary variable as before and using a new parameter  $\delta$ , we obtain:

$$\begin{aligned} &\Pr(r_{i,j} | P_{u_i}^{true}, F_{o_j}^{true}, C_{o_j}^{true}) \\ &= \left( \Pr(r_{i,j} | P_{u_i}^{true}) \vartheta + \Pr(r_{i,j} | F_{o_j}^{true}) (1 - \vartheta) \right) (1 - \delta) + \Pr(r_{i,j} | C_{o_j}^{true}) \delta \end{aligned} \quad (16)$$

The argument that maximizes the above expression indicates the rating that user  $u_i$  is most likely to assign to the object  $o_j$ .

### 3.3 Score Combination

So far, in most of the previously developed systems, the merging of information sources is carried out by a naïve linear combination of numerical scores resulting from each prediction technique individually in order to give a single prediction. In our case, where we combine scores from three different sources, the prediction is calculated as follows:

$$P_{i,j} = \left( \begin{array}{l} \arg \max_r \left( \Pr(r_{i,j} | P_{u_i}^{true}) \right) \vartheta + \\ \arg \max_r \left( \Pr(r_{i,j} | F_{o_j}^{true}) \right) (1 - \vartheta) \end{array} \right) (1 - \delta) + \arg \max_r \left( \Pr(r_{i,j} | C_{o_j}^{true}) \right) \delta \quad (17)$$

## 4 Experimental Evaluation

We carried out our experiments using the MovieLens dataset, taken from a research recommendation site being maintained by the GroupLens project [13]. The MovieLens dataset contains 100.000 ratings, scaling from 0 to 5, derived from 943 users on 1682 movie titles (items) where each user has rated at least 20 movies. We first carried out some experiments to tune the weighting parameters  $\vartheta$  and  $\delta$  and then using selected values of them we tested our framework along with other algorithms. Metrics

used to evaluate the quality of recommendations are the Mean Absolute Error (MAE) and the F1, which is the harmonic mean of precision and recall.

4.1 Configuration

Parameter  $\vartheta$  adjusts the balance between PD and FD prediction techniques (we denote this combination with PFD), while parameter  $\delta$  adjusts the balance between PFD and CD (henceforth PFCD). We vary each user’s number of observed as well as each item’s number of raters to find the best possible configurations for our combination schemes.

First, we use the MAE metric to examine the sensitivity of both schemes over  $\vartheta$ . For this purpose, we set the value of  $\delta$  to zero. Then, we vary  $\vartheta$  from zero (pure FD) to one (pure PD). We test over user-sparsity 5 and 20, and item sparsity less than 5 and less than 20. Regarding the sum-rule scheme, as Fig. 1a shows, the best results are achieved for values of  $\vartheta$  between 0.3 and 0.7. For this range of values the prediction accuracy can be improved up to the 8% of the technique with the best accuracy when it is used

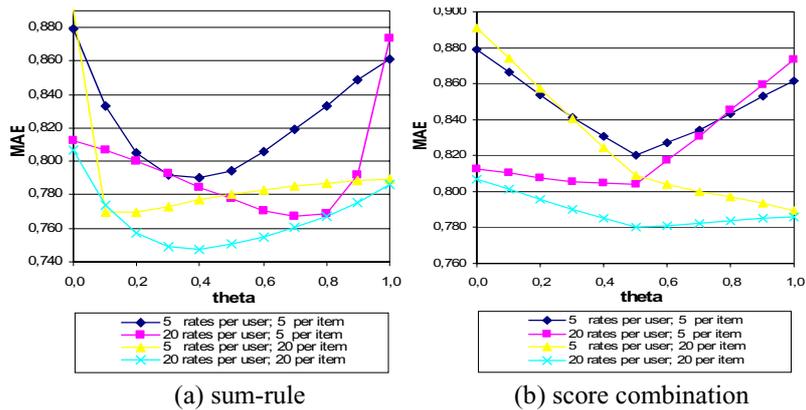


Fig. 1. Impact of parameter  $\vartheta$  in sum-rule and score combination schemes

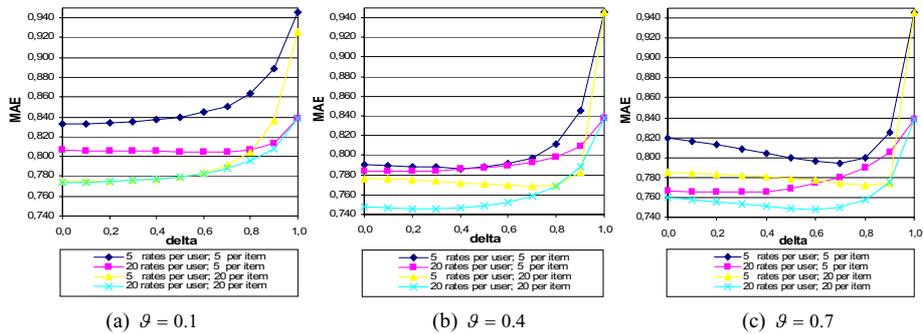


Fig. 2. Impact of parameter  $\delta$  for different values of  $\vartheta$  in sum-rule

individually. In Fig. 1b, for score combination scheme, we obtain optimum results for values of  $\vartheta$  greater than 0.5 except the third configuration since no combination of PD and FD seems to give a better MAE than PD itself.

In Fig. 2, for the sum-rule scheme, we assign to  $\vartheta$  values 0.1 (denoted with PFCDs\_1), 0.4 (PFCDs\_2) and 0.7 (PFCDs\_3) and test the sensitivity with respect to the parameter  $\delta$ . Using the same configurations over user and item sparsity we vary  $\delta$  from zero (pure memory-based) to one (pure content-based). Figs. 2a and 2b, for PFCDs\_1 and PFCDs\_2 respectively, show no clear improvement of MAE over  $\delta$ . As for PFCDs\_3 (Fig. 2c), we obtain the optimum results for values of  $\delta$  between 0.2 and 0.8, which improve MAE almost by 4%. Based on the above observations, we tune  $\delta$  to 0.1 in PFCDs\_1, 0.7 in PFCDs\_2 and 0.6 in PFCDs\_3 to further experiment with their overall performance.

For the score combination schemes we set the value of  $\vartheta$  to be equal to 0.5 (PFCDn\_1) and 0.8 (PFCDn\_2) and test the sensitivity regarding parameter  $\delta$ . Using the same configurations over user and item sparsity we vary  $\delta$  from zero (pure memory-based) to one (pure content-based). As shown in Fig. 3, using PFCDn\_1 and PFCDn\_2 in the recommendation process does not seem to improve the quality of prediction. Some exceptions are the sparsity configurations in which only 5 item votes are kept throughout the recommendation process (first and second configuration). After these observations we set  $\delta$  to 0.3 in PFCDn\_1 and 0.4 in PFCDn\_2.

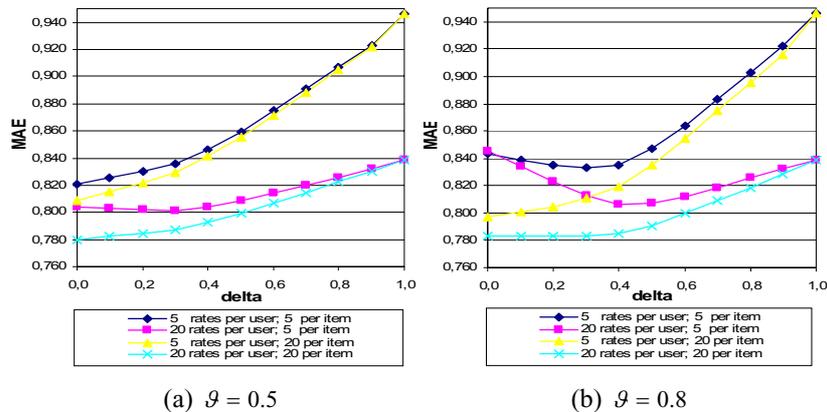


Fig. 3. Impact of parameter  $\delta$  for different values of  $\vartheta$  in score combination

## 4.2 Overall Performance

In this section, we randomly select parts of the training data and test the previous configurations along with the product-rule (we denote with PFCDp) and, moreover, with other memory-based algorithms as are the user (UBPCC) and item (IBPCC) Pearson correlation coefficient in terms of overall sparsity. The results in Table 1 indicate the superiority of the purely probabilistic schemes against the naïve score combination schemes with respect to the quality of the prediction accuracy (MAE and F1). More specifically, the purely probabilistic schemes can provide better results, up

to 10%. However, this conclusion does not stand for every pair of parameters  $\vartheta$  and  $\delta$  - e.g., as shown in Table 1, score combination scheme PFCDn\_2 outperformed the purely probabilistic scheme PFCDs\_1 in terms of F1.

The results, finally, prove our initial assumption about KNN algorithms; in particular, they require the existence of common items between users. This is why the F1 metric has a decreased value for the methods UBPC and IBPC.

**Table 1.** MAE and F1 over different sparsity levels

	MAE					F1				
	20%	40%	60%	80%	100%	20%	40%	60%	80%	100%
PFCDs_1	0,859	0,817	0,789	0,783	0,781	0,631	0,669	0,681	0,692	0,695
PFCDs_2	0,821	0,788	0,778	0,767	0,764	0,673	0,704	0,715	0,722	0,723
PFCDs_3	0,807	0,776	0,765	0,761	0,758	0,683	0,713	0,721	0,727	0,729
PFCDn_1	0,918	0,866	0,841	0,828	0,823	0,612	0,653	0,656	0,676	0,681
PFCDn_2	0,878	0,827	0,810	0,793	0,790	0,647	0,683	0,693	0,705	0,711
PFCDp	0,812	0,780	0,771	0,760	0,758	0,677	0,709	0,719	0,725	0,727
PD	0,884	0,838	0,815	0,806	0,797	0,641	0,681	0,693	0,701	0,707
UBPCC	1,022	0,904	0,866	0,844	0,830	0,159	0,419	0,483	0,505	0,511
IBPCC	0,999	0,895	0,852	0,836	0,824	0,181	0,407	0,473	0,495	0,507

## 5 Discussion

In this paper, we proposed the use of the two basic combination schemes from the theory of probabilities in order to overcome accuracy issues of the RS. Results showed that purely probabilistic schemes provide better quality results than naïve linear weighting of scores derived from all techniques individually. However, the results are very sensitive to the tuning parameters - it is not clear at all how to set theta and delta in a robust way. Moreover, it is worth noticing that in most cases the product-rule which requires no tuning was outperformed slightly by the sum-rule in its best configuration (i.e. PFCDs\_3). The main reason is the sensitivity in errors, which is intense in the latter case due to the factorization of prediction techniques; i.e., independence of the techniques does not always hold. For more details we refer to [8]. It is also important to notice that the combination of more than two prediction techniques does not always improve the output. Since a RS consists of a voting system we believe that this observation is related to the Arrow's Paradox [1]. Our future study will also take into account this issue.

## References

1. Arrow, K.J.: Social Choice and Individual Values. Ph.D. Thesis, J. Wiley, NY (1963)
2. Billsus, D., Pazzani, M.: User Modeling for Adaptive News Access. *User-Modeling and User-Adapted Interaction* 10(2-3), 147–180 (2000)
3. Breese, J.S., Heckerman, D., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI 1998)*, July 1998, pp. 43–52 (1998)

4. Burke, R.: Hybrid Recommender Systems: Survey and Experiment. *User Modeling and User-Adapted Interaction* 12(4), 331–370 (2002)
5. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M.: Combining Content-Based and Collaborative Filters in an Online Newspaper. In: *SIGIR 1999 Workshop on Recommender Systems: Algorithms and Evaluation*, Berkeley, CA (1999)
6. Deshpande, M., Karypis, G.: Item-based Top-N Recommendation Algorithms. *ACM Trans. Inf. Syst.* 22(1), 143–177 (2004)
7. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An Algorithmic Framework for Performing Collaborative Filtering. In: *Proc. of SIGIR* (1999)
8. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On Combining Classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.* 20(3), 226–239 (1998)
9. Linden, G., Smith, B., Smith, J.X.: Amazon.com Recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 76–80 (January/February 2003)
10. Papagelis, M., Plexousakis, D., Kutsuras, T.: A Method for Alleviating the Sparsity Problem in Collaborative Filtering Using Trust Inferences. In: *Proceedings of the 3rd International Conference on Trust Management* (2005)
11. Pazzani, M.J.: A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review* 13(5/6), 393–408 (1999)
12. Pennock, D.M., Horvitz, E.: Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-based Approach. In: *Proceedings of UAI* (2000)
13. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: *CSCW 1994: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, Chantilly, North Carolina, United States, pp. 175–186. ACM Press, New York (1994)
14. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based Collaborative Filtering Recommendation Algorithms. In: *WWW 2001: Proceedings of the 10th International Conference on World Wide Web*, pp. 285–295. ACM Press, Hong Kong (2001)
15. Tran, T., Cohen, R.: Hybrid Recommender Systems for Electronic Commerce. In: *Knowledge-Based Electronic Markets, Papers from the AAAI Workshop*, AAAI Technical Report WS-00-04, pp. 78–83. AAAI Press, Menlo Park (2000)
16. Wang, J., de Vries, A.P., Reinders, M.J.: A User-Item Relevance Model for log-based Collaborative Filtering. In: Lalmas, M., MacFarlane, A., Rüger, S.M., Tombros, A., Tsirikia, T., Yavlinsky, A. (eds.) *ECIR 2006*. LNCS, vol. 3936, Springer, Heidelberg (2006)
17. Wang, J., de Vries, A.P., Reinders, M.J.: Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion. In: *Proceedings of SIGIR* (2006)