Measuring Predictive Capability in Collaborative Filtering

Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete,
Miguel A. Rueda-Morales
Dpt. of Computer Science and Artificial Intelligence
CITIC-UGR, University of Granada
18071 Granada, Spain
(lci,jmfluna,jhg,mrueda)@decsai.ugr.es

ABSTRACT

This paper presents a new memory-based approach to Collaborative Filtering where the neighbors of the active user will be selected taking into account their predictive capability. Our hypothesis is that if a user was good at predicting the past ratings, then his/her predictions will be also helpful to recommend ratings in the future. The predictive capability of a user will be measured using two different criteria: The first one which is based on the likelihood of the active user's rating and the second one tries to minimize the error obtained using his/her predictions. We show our experimental results using standard data sets.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: General

General Terms

Algorithms, Theory

Keywords

Probabilistic Reasoning, Collaborative Filtering, Recommender System

1. INTRODUCTION

The usual formulation of the recommending problem is to predict how an active user might rate an unseen item. Recommender Systems are often based on *Collaborative filtering* (CF) [6], which relies only on past user behavior. These systems attempt to identify groups of people with similar tastes to those of the user and recommend items that they have liked. In this paper we focus on memory-based approaches that uses the entire rating matrix to make predictions (in contrast to model-based approach that learns an abstraction [4]). The advantage of memory-base approach with respect to model-based is that a fewer number of parameters have to be tuned [8], however they have some problems with the sparsity of the data. Nevertheless, memory-based approaches have reached a great popularity because they are simple and intuitive on a conceptual level being also sufficient for many real-world problems [3, 7, 5, 8].

Different views of the rating matrix leads to two different types of memory-based approaches: user-based methods where the predictions are generated based on those ratings given to the target item by similar users [3], and item-based methods where the predictions are generated considering those ratings given by the active user to similar items [7]. Focusing on user-based model, a critical step is to identify users that are similar to some active user, A. The way in which this similarity is computed can have a significant impact on the performance of the system. Studies and real-world implementations so far have relied on traditional vector similarity measures (say Pearson's correlation in different formulations, cosine, Spearman's Rank, etc. [3, 1, 8]) or using a probabilistic framework [8].

The objective in this paper is to create a memory-based CF model trying to better predict the probability of the alternative ratings. This model will be an example of Predictive modeling [2]. Our hypothesis is that if a user was good at predicting the past ratings, then his/her predictions shall be also helpful to recommend ratings in the future. Two different alternatives to measure the predictive capability of a user will be considered: The first one that considers how probable is a given user acting as predictor and the second one which tries to minimize the error obtained using his/her predictions.

In order to recommend a rating for the target item, I_k, we propose to select the best-N users (those with highest predictive capability) among the set of users who previously rated this item. Then, in a second step, the rating for the target item has to be predicted. Usually, this rating is computed by averaging the (weighted) known ratings given by those similar users to I_k [8]. Taking into account our predictive purposes in this paper we will also explore a different alternative to aggregate the information. Briefly, we compute a probability distribution over the candidate ratings, and then this distribution is used to perform the final prediction.

In the process above, the information about the target item only acts as a filter (we focus on those users who rated this item previously). Then, for each user who passes this filter, we measure whether the predictions over *all* the A's past ratings are good or not. In this paper we will also explore a different approach. Let us consider the following example. If I wish to obtain a prediction for the movie "Monsters vs Aliens", the quality of a user can be measured

RecSys'09, October 23–25, 2009, New York, New York, USA.

by considering his/her predictive capability for those movies which are similar to the target one. In some sense we are giving some opportunities to those users who might do well predicting animation movies but their predictions are not good enough if we consider others genres.

This paper is organized in the following way: Next section presents an overview of the predictive model. The different selecting criteria will be studied in Section 3. How it can be used item's similarities is presented in Section 4. Section 5 describes the proposed experimentation and Section 6 presents the final conclusions.

2. PREDICTIVE MODEL

Firstly, we shall introduce some notation: Let A be the active user and let U (or U_i) be any other user in the system. We denote by D_a (respectively D_u) the set of ratings given by A (respectively U). We also use r_a^j to denote the rating given by user A to the j^{th}-item ($r_a^j = 0$ when user A does not rate this item). Given the target item I_k, we denote by C_k those users who rated this item. Finally, the set of valid ratings is denoted by S.

In order to build a predictive model, we have to look for those elements that are likely to influence the active user's future ratings. In a CF framework these elements are those ratings given by the rest of the users. Briefly (see the algorithm in Table 1), our predictive model will select the best-N users (sorted using their predictive capability over the past ratings) among those users who rated the target item, C_k. In order to measure the predictive capability of a given user, $U \in C_k$, we shall consider two different alternatives depending on whether U is a predictor for the active user (denoted by $\mathcal{H}_1$) or not (denoted by $\mathcal{H}_2$)

- $\mathcal{H}_1$. In this case, we are considering that given the ratings in D_u, the active user might rate I_k with probability $P(r_a^k|\mathcal{H}_1, D_u), \forall r_a^k \in S$.
- $\mathcal{H}_2$. In this case, we assume that the ratings of the active user can be predicted using a probability $P(r_a^k)$ which does not depend on those ratings given by U, i.e. $P(r_a^k) = P(r_a^k|\mathcal{H}_2, D_u), \forall r_a^k \in S$.

The particular way used to measure the predictive power of an user will be discussed in Section 3.

Once the set of best-N predictors have been selected from C_k, their individual predictions (in terms of probability distributions over ratings, $P(r_a^k|\mathcal{H}_1, D_u), \forall r_a^k \in S$) have to be aggregated (step 2 in Table 1). Since we know the rating given by each user $U_j \in C_k$ to the target item I_k, say r_{uj}^k, we will use the following conditional independence assumption:

A1 Given r_{uj}^k, the predictive probabilities associated to U expressing our belief about how likely the active user could rate I_k with r_a^k will not depend on those ratings given by U to the rest of the items in D_u, i.e. $P(r_a^k|\mathcal{H}_1, D_u) = P(r_a^k|\mathcal{H}_1, r_{uj}^k)$.

In this paper, the posterior probability distribution over the candidate ratings will be obtained by means of a linear combination of the individual predictive probabilities, i.e.,

$$P(r_a^k = s) = \sum_{j=1}^{N} P(r_a^k = s|\mathcal{H}_1, r_{uj}^k)\alpha_{j,a}, \quad \forall s \in S \qquad (1)$$

Inputs: A active user, I_k target item Output: $\hat{r}_a^k$ predicted rating.
1. Neighbors Selection 1.1 For each $U_i \in C_k$ compute its Predictive Capability. 1.2 Select the best-N predictors 2. Predictive Process 2.1 Combine the predictions to obtain $P(r_a^k = s), \forall s$. 2.2 Return the median rating.

Table 1: Predictive Model

being $\alpha_{j,a}$ normalized weights giving more strength to most valuable users. In Section 5 we will discuss how these values might be computed.

Since the objective is the prediction of the rating that the active user should give to I_k, $\hat{r}_a^k$ a final decision becomes necessary. There are several methods for computing this prediction from a probability distribution over ratings. For example, we can use the most probable, the average or the median rating. In this paper we will use the median prediction since it minimizes the mean absolute error (an standard metric used in CF which will be also used in our experimentation). Therefore,

$$\hat{r}_a^k = \{r|P(r_a^k < r) \leq 0.5, P(r_a^k > r) \geq 0.5\}. \qquad (2)$$

We would like to note that the way in which we compute the predicted rating represents another difference between our approach and the ones in [3, 8]. In these models the prediction is computed based on a weighted aggregation of those ratings given by the selected neighbors whereas our approach uses a weighted aggregation of the individual predictive probabilities.

3. COMPUTING PREDICTIVE CAPABILITY

In this paper we are going to explore two different alternatives to measure the predictive capability of a given user U: The first one that considers how probable is that U acts as a predictor and the second one, which tries to minimize the error obtained using his/her predictions.

3.1 Is U a good predictor?

Our objective is to measure how probable is *"U is a predictor of A"* given that we know their past ratings, D_a and D_u, i.e. $P(\mathcal{H}_1|D_a, D_u)$. Considering the Bayes' theorem we have that

$$P(\mathcal{H}_1|D_a, D_u) = \frac{P(D_a|\mathcal{H}_1, D_u)P(\mathcal{H}_1|D_u)}{P(D_a|D_u)}, \qquad (3)$$

with

$$P(D_a|D_u) = P(D_a, \mathcal{H}_1|D_u) + P(D_a, \mathcal{H}_2|D_u)$$
$$= P(D_a|\mathcal{H}_1, D_u)P(\mathcal{H}_1|D_u) + P(D_a|\mathcal{H}_2, D_u)P(\mathcal{H}_2|D_u).$$

In this equation $P(\mathcal{H}_i|D_u)$ represents the subjective prior probability over our hypothesis space, which expresses how plausible we thought $\mathcal{H}_i$ was before the D_a data arrived. In other words, given that we know the set of ratings given by U, what are our beliefs about "U is a predictor for the active user". In this paper we shall assume that the two alternatives, $\mathcal{H}_1$ and $\mathcal{H}_2$, are equally probable, although some other approaches might be considered. For example we could consider that these values might depend on the number of items

rated by U, the rating variability of U (an user who always rates with almost the same value does not help in the predictions) or depending on some social factor controlling how influential a given user is.

The term $P(D_a|\mathcal{H}_2, D_u)$ represents the probability of the ratings D_a when it is assumed that U is not a predictor for A, i.e., the ratings of A do not depend on those ratings given by U. Therefore, it is natural to consider that $P(D_a|\mathcal{H}_2, D_u)$ is equal to $P(D_a)$.

In order to rank the users according to $P(\mathcal{H}_1|D_a, D_u)$, we will use the following proposition, which states that the same ranking will be obtained using the likelihood $P(D_a|\mathcal{H}_1, D_u)$ (we essentially ignore those terms which do not affect the ranking).

Proposition: *Given any two users U and U', if $P(D_a|\mathcal{H}'_1, D'_u) < P(D_a|\mathcal{H}_1, D_u)$ then $P(\mathcal{H}'_1|D_a, D'_u) < P(\mathcal{H}_1|D_a, D_u)$.*

So, the problem reduces to compute $P(D_a|\mathcal{H}_1, D_u)$. In this sense, let $D_a = \{r_a^1, \ldots, r_a^t\}$ be the set of t ratings given by A. In order to compute these values we will consider the following assumption:

A2 *The active user's ratings are (marginally) independent:* By means of this assumption we have that

$$P(D_a|\mathcal{H}_1, D_u) = \prod_{i=1}^{t} P(r_a^i|\mathcal{H}_1, D_u).$$

Given that we know how the user U rated each item I_i in D_a, r_u^i (if U did not rate the item, it is considered that $r_u^i = 0$, i.e. we use the null value as the given rating), and considering the conditional independence assumption, A1, we have that $P(r_a^i|\mathcal{H}_1, D_u) = P(r_a^i|\mathcal{H}_1, r_u^i)$. These probabilities shall be estimated by relative counts over the set of items rated by either A or U, i.e. $D_a \cup D_u$, using Laplace smoothing.

Therefore, the likelihood of the active user's ratings shall be computed as (we will work with the natural log)

$$\log P(D_a|\mathcal{H}_1, D_u) = \sum_{i=1}^{t} \log P(r_a^i|\mathcal{H}_1, r_u^i).$$

3.2 Minimizing the error in the predictions

In this paper we shall study another alternative for measuring the predictive power of a user (step 1 in the algorithm in Table 1). The idea will be to measure for each user U who had rated the target item I_k, U in C_k, the possible loss associated with those predictions obtained when estimating the active user's past ratings. Particularly, and in order to measure the difference between the estimated rating and the true rating, EL, we propose to use the mean absolute error (MAE) over A's ratings, $D_a = \{r_a^1 \ldots, r_a^t\}$, i.e.

$$EL(A, U) = \frac{\sum_{i=1}^{t} |\hat{r}_{a,u}^i - r_a^i|}{t}.$$

In this equation, the term $\hat{r}_{a,u}^i$ denotes the estimated rating for an item I_i in the case of being U a predictor of A. Therefore, in these predictions the predictive probabilities of the user U, $P(r_a^i|\mathcal{H}_1, D_u)$ have to be used. Following the ideas of the proposed model and considering the assumption A1, the estimated rating is the median of the individual predictive probabilities, i.e.

$$\hat{r}_{a,u}^i = \{r|P(r_a^i < r|\mathcal{H}_1, r_u^i)) \leq 0.5, P(r_a^k > r|\mathcal{H}_1, r_u^i)) \geq 0.5\}.$$

Then, the best N users, i.e. those users achieving the lowest expected loss, $EL(A, U)$, will be selected as neighbors of A.

4. CONSIDERING ITEM SIMILARITIES

The above metrics try to capture how good is a user U when predicting A's ratings. In order to compute these metrics we have considered the influences of the user U over *all* the past ratings given by A, $D_a = \{r_a^1, \ldots, r_a^t\}$. In this process, the information about the item which is going to be recommended I_k only acts as a filter: The required predictive measures are computed only for those users who rated I_k previously, i.e. $U \in C_k$ (step 1.1 in Table 1).

These computations might require a time[1] in the order $(O(t \times |C_k|))$ which could be prohibitive in real online applications. In a steady situation, where it is not common to include new ratings, these computations can be performed offline, but this is not the usual case in a recommending framework. This problem is common to all the memory-based approaches for CF. Traditionally these methods select (a priori) more users, although not all of them have given a rating for the items that we wish to predict. As consequence, the predictions obtained are not necessary based on N ratings.

In this paper we will study a different approach to reduce the efforts necessary to compute the predictive power of an user U. The idea is to measure the quality based on the predictions obtained for those items which are **similar** to the target item I_k. Thus, if D_a^k ($D_a^k \subset D_a$) is the set of ratings given by A to the M items most similar to I_k, our hypothesis is that if a user did well predicting the ratings in D_a^k then his/her predictions could be helpful to predict a rating for I_k. In order to measure the similarity between items we shall use the adjusted cosine between item's ratings [7]. In some way we are combining both user-based and item-based approaches, combination which has been shown beneficial in other models [8], where the estimated rating is combination of the predicted rating using the two approaches.

5. EXPERIMENTATION

In order to evaluate our approach we used MovieLens (ML) and Yahoo! Movies (Yh) [9] data sets: MovieLens contains ratings of 1682 movies rated by 943 users. We have used the original training and test data sets -with 80,000 and 20,000 ratings. On the other hand, Yahoo! Movies contains ratings of 11,915 movies rated by 7,642 users also divided into training and test sets with 211,231 and 10,136 ratings, respectively. As accuracy criterion we have used the MAE metric that measures how close the predictions are to the original ratings. We have used 5-fold cross validation protocol to evaluate the results, also and in order to select the best-N users it has been considered all the users in C_k with N taking the values 10, 20, 30, 50 and 75. Also, when considering item similarities we have used the $M = 10$, i.e. the 10 most similar items.

We have also taken into account two different baselines: B is the model in [3] where the best N neighbors have been selected using Pearson Correlation between users. We wish to note that although this model is simple its performance remains competitive. On the other hand, we have used a model-based CF approach [4] related to the aspect model

[1]We are not considering here the time needed to sort (step 1.2 in Table 1) the predictive probabilities.

MAE's using MovieLens					
	10	20	30	50	75
B	0.7448	0.7344	**0.7331**	0.7339	0.7355
$H\alpha^P$	0.7303	0.7244	**0.7235**	0.7246	0.7246
$H\alpha^H$	0.7470	0.7427	0.7422	0.7418	0.7412
$E\alpha^P$	0.7303	0.7279	**0.7275**	0.7278	0.7281
$E\alpha^E$	0.7417	0.7427	0.7427	0.7437	0.7432
Using 10 most similar items					
$H\alpha^P$	0.7283	0.7215	**0.7208**	0.7216	0.7233
$H\alpha^H$	0.7530	0.7471	0.7452	0.7489	0.7466
$E\alpha^P$	0.7432	0.7350	0.7324	0.7289	**0.7283**
$E\alpha^E$	0.7480	0.7430	0.7446	0.7421	0.7434
MAE's using Yahoo! Movies					
	10	20	30	50	75
B	0.7504	0.7390	0.7351	0.7324	**0.7319**
$H\alpha^P$	0.7331	0.7317	0.7303	**0.7287**	0.7293
$H\alpha^H$	0.7644	0.7561	0.7568	0.7597	0.7588
$E\alpha^P$	0.6833	**0.6771**	0.6772	0.6796	0.6817
$E\alpha^E$	0.7623	0.7551	0.7556	0.7614	0.7650
Using 10 most similar items					
$H\alpha^P$	0.7318	0.7313	0.7296	**0.7285**	0.7291
$H\alpha^H$	0.8578	0.8442	0.8302	0.8195	0.8092
$E\alpha^P$	0.7369	0.7175	0.7056	0.7022	**0.6997**
$E\alpha^E$	0.7990	0.7833	0.7768	0.7781	0.7806

Table 2: MAE's using MovieLens and Yahoo! Movie

for probabilistic semantic analysis[2]. Particularly, the MAE values obtained using 10 "user types" are 0.7425 and 0.7458 with MovieLens and Yahoo! movies, respectively.

The two parameters evaluated in the experimentation are, on the one hand, the criterion used to select the neighborhood (see Section 3). In this case, we use H to denote that the best users have been selected using their predictive probabilities and E to denote that we are trying to minimize the error in the predictions. With respect to the weights $\alpha_{j,a}$ used in Equation 1 representing the strength given to the j^{th}-user we shall explore three different approaches: Normalized Pearson correlation coefficient (α^P) over users ratings, normalized likelihoods $P(D_a|H_1, D_u)$ (α^H) and normalized error loss $EL(A, U)$ (α^E). Table 2 shows the result obtained in our experimentation. Also, an in order to study the scalability of the approach, we have performed a brief experimentation with 1 Million Movielens data set. Table 3 shows the results obtained after considered 90% training and 10% test. In this case we have fixed to 30 the user's neighborhood.

From these data we can conclude that our proposals are competitive (our best results outperforms those obtained with the baselines). With respect to the criteria used to select the active user neighborhood the minimization of the loss function shows a good performance with all data sets, although the results are remarkable for Yahoo! data and 1M Movielens). Also, the way in which the α weights have been computed have an important role in the accuracy of the predictions. Surprisingly, the use of Pearson coefficient, α^P, stand out in all the experiments. We believe that this is because our "naive" approach to normalize the other weights, so a further studies become necessary. Finally, the use of items similarities have been proved beneficial with MovieLens data set (we could obtain similar results with less computational time). The results obtained with Yahoo! data and 1M Movielens are not conclusive. We guess that the differences in performance between data sets are due to the difficult of finding similarities between items. Nevertheless, we have to say that the predictions can be obtained with less computational time.

[2]This model considers a latent variable which can be interpreted as a "user type". Briefly, in this model a user is seen as a distribution over user types and for each user type we can obtain a distribution over the pairs item-rating.

	B	$H\alpha^P$	$H\alpha^H$	$E\alpha^P$	$E\alpha^E$
NO-IS:	0.7082	0.6901	0.7925	**0.6855**	0.6972
IS:		0.7159	0.7203	**0.7045**	0.7246

Table 3: MAE's using 1 Million MovieLens Data Set with and without the use of items-similarity (IS and NO-IS, respectively.)

6. CONCLUSIONS

In this paper, we show the feasibility of building a memory-based CF system that considers those users having greater impact in the (past) ratings of the active user. The performance is improved by taking into account how these users influence in the ratings of items similar to the target one. As future work we plan to study the use of content information in order to determine items similarities and also the use of a variable number of neighbors in the recommendation process

Acknowledgements: This work has been jointly supported by the Spanish Ministerio de Educación y Ciencia, and the research program Consolider Ingenio 2010, under projects TIN2005-02516, TIN2008-06566.C04.01 and CSD2007-00018, respectively.

7. REFERENCES

[1] H.J. Ahn. *A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem.* Information Sciences 178:37-51, 2008.

[2] S. Geisser (1993) Predictive Inference: An Introduction. New York: Chapman & Hall.

[3] J. L. Herlocker, J. A. Konstan, A. Borchers, J. Riedl. *An algorithmic framework for performing collaborative filtering.* Proc. ACM SIGIR 99, pp. 230–237. 1999

[4] T. Hofmann. *Latent semantic models for collaborative filtering.* ACM TOIS V22 (1) pp 89–115. 2004

[5] G. Linden, B. Smith, J. York. *Amazon.com Recommendations: Item-to-Item collaborative filtering* IEEE Internet Computing, pp. 76-80. 2003.

[6] P. Reskick, H.R. Varian. *Recommender systems.* Communications of the ACM, 40(3):56–58, 1997.

[7] B. Sarwar, G. Karypis, J. Konstan, J. Reidl. Item based collaborative filtering recommendation algorithms, Proc. 10th Int. Conference on WWW, pp. 285-295, 2001,

[8] J.Wang, A.P. de Vries, M.Reinders *Unified Relevance Models for Rating Prediction in Collaborative Filtering* ACM TOIS, 26:3. Article 16. 2008

[9] *Y*ahoo! Webscope dataset ydata-ymovies-user-movie-ratings-v1.0 http://research.yahoo.com/Academic_Relations