



Repositorio Institucional de la Universidad Autónoma de Madrid

<https://repositorio.uam.es>

Esta es la **versión de autor** de la comunicación de congreso publicada en:
This is an **author produced version** of a paper published in:

CAMRa '11 Proceedings of the 2nd Challenge on Context-Aware Movie
Recommendation, ACM, 2011. 29-35

DOI: <http://dx.doi.org/10.1145/2096112.2096118>

Copyright: © 2011 ACM

El acceso a la versión del editor puede requerir la suscripción del recurso
Access to the published version may require subscription

Temporal Rating Habits: A Valuable Tool for Rating Discrimination

Pedro G. Campos^{1,2}
pedro.campos@uam.es

Fernando Díez¹
fernando.diez@uam.es

Alejandro Bellogín¹
alejandro.bellogin@uam.es

¹Universidad Autónoma de Madrid
Francisco Tomás y Valiente 11
28049, Madrid, Spain

²Universidad del Bío-Bío
Av. Collao 1202
4081112, Concepción, Chile

ABSTRACT

In this paper, we describe the experiments conducted by the *Information Retrieval Group* at the Universidad Autónoma de Madrid (Spain) to tackle the Identifying Ratings (track 2) task of the CAMRa 2011 Challenge. The experiments performed include time-frequency probabilistic strategies, heuristic collaborative filtering (CF) and a model-based CF approach. Results show that probabilistic classifiers based on temporal behavior of users have better performance than traditional recommendation-based strategies, thus reflecting that temporal information is a valuable source for the identification or discrimination of user ratings.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Information Filtering, Retrieval Models, Selection Process*; I.5.1 [Pattern recognition]: Models

General Terms

Algorithms, Performance, Experimentation

Keywords

Context-Aware Recommender Systems, Movie Recommendation, Probability Models

1. INTRODUCTION

Information about context can help improving personalization-related tasks [1]. The Challenge on Context-aware Movie Recommendation 2011 (CAMRa2011) provides an interesting opportunity to test recommendation approaches on real data. We focus on the *Identifying Ratings* track, which consists of determining which members of a *household* made some specific “unidentified” ratings, once information about which users belong to each household is known, along with the movie ratings assigned by the households. In this case,

there are two dimensions of contextual information. On the one hand, household information, which may allow us to take advantage of knowing the existence of a relationship among some users (although the actual relation remains unknown), and on the other hand, temporal data, since each rating has an associated timestamp, which allows to track users’ concept drift. Nonetheless, other interesting information which has been used previously in different recommendation strategies such as movies features (*title, genre*), user demographics and other social relationships, are not available in this challenge. This fact makes more difficult to define relations between each item in question and the users to be allocated.

Considering the above issues mentioned, we conducted a series of experiments with different models, in order to better predict to whom each “unidentified” rating belongs to, which we describe in this work. The remainder of this paper is structured as follows. Section 2 describes the main characteristics of the available data for the competition. Section 3 presents a brief review of models that could be used, considering information available in the dataset. Section 4 details the models used for making the predictions. Section 5 presents the results obtained, along with the evaluation methodology followed and required by the challenge. We finalize with some concluding remarks and devised additional approaches to experiment in Section 6.

2. DATASET ANALYSIS

2.1 General Description

CAMRa 2011’s MoviePilot Dataset consists of a training set of 4,536,891 timestamped ratings from 171,670 users on 23,974 items on a timespan from July 11, 2009 up to July 12, 2010, and two test sets (one for each competition track): track 1 containing 4482 ratings from 594 users on 811 items on a timespan from July 15, 2009 up to July 10, 2010 and track 2 containing 5450 timestamped ratings from 592 users on 1706 items on a timespan from July 13, 2009 up to July 11, 2010. Since we focus only on track 2, from now on we only analyze data related with that track.

Figure 1 shows the rating, community and catalog growth of training data (upper side) and testing data for the track 2 (lower side) through time. It may be seen that data growth follows a similar proportion on both data splits. It is also available, for some users, information about which household a user belongs to. Table 1 shows the size distribution of households in the dataset. 2-sized households represent

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CAMRa 2011 October 27, 2011, Chicago, IL USA

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$10.00.

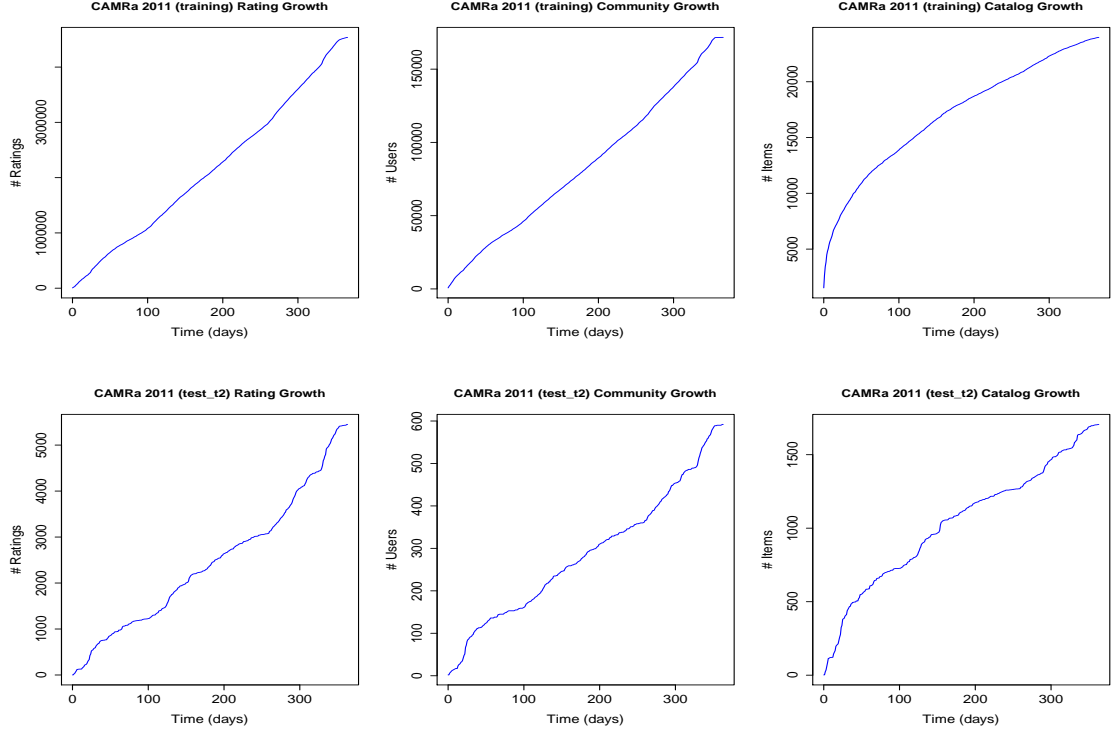


Figure 1: Training (upper) and testing data growth through time

Table 1: Households' size frequencies

Size	2	3	4	All
Frequency	272	14	4	290

the 93.8% of all the households, whilst 3-sized and 4-sized households represent the 4.8% and 1.4% respectively. Note that, although there are 602 users which are members of some household (they appear in the training set), only 592 users have data in the test set for this track.

2.2 Frequency-based Analysis

Taking into account that we do not know whether the household's relationships correspond to friends, siblings, couples, etc., and that no other information is provided, we focused our analysis on temporal trends which may help us on completing the task at hand. We performed a descriptive study of the given characteristics on the training data and we observed a phenomenon repeated in several of the users belonging to different households. In Figure 2, it is shown the rating hour *probability mass functions* (PMFs from now on) of two users in the first household. We can observe here that there is a clear disparity between the hours employed by each of the household's members for rating movies. The user u40426 has a probability near 1 (0.93) to rate movies in the period from 18:00 to 19:00. On the contrary, user u311738 rates movies starting at 20:00 and later on, that is, mostly by night. Similar circumstances to the one described above are repeated along the data set, suggesting time-aware strategies might be useful.

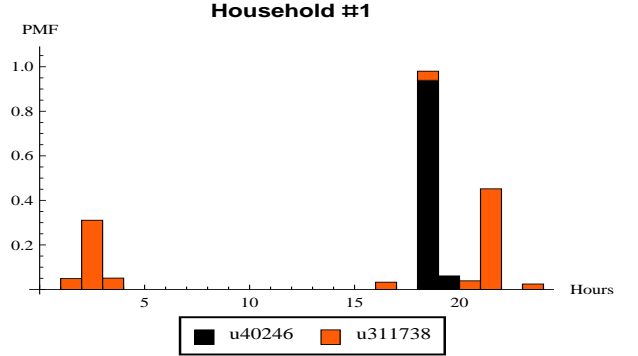


Figure 2: PMF of the user rating hours

When analyzing the rating date from each user, it is also possible to detect some interesting facts. Figure 3 shows how many ratings are made by users through time. The left frame shows that the mean user rating window size (i.e. timespan at which users make ratings) is very small (just a few days). The center and right frames also show that the vast majority of ratings are incorporated during the first days of participation of a user. Considering that users start their participation on different days, this information can be helpful in our task. We also noted that there are differences on which day of the week each user rates movies, although, for the sake of clarity, we preferred to leave those figures out of this paper.

The analysis of the raw rating value frequency alone also

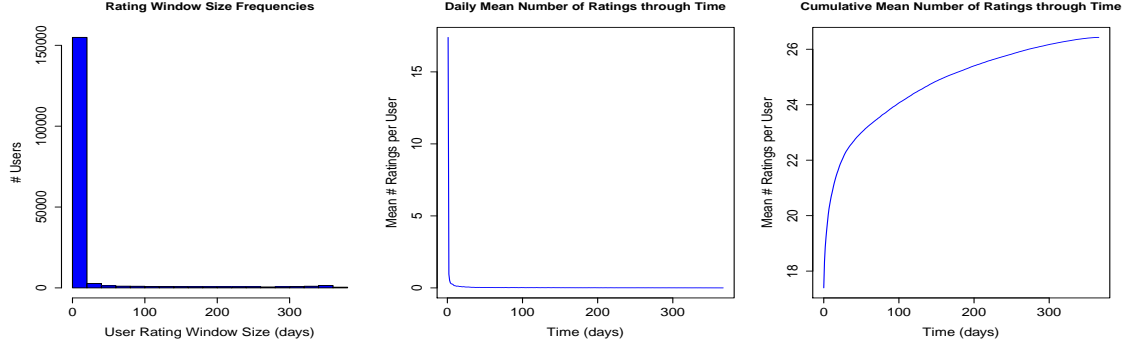


Figure 3: User’s rating frequencies through time (from left to right, rating window size, daily and cumulative number of ratings through time)

gives us some clues about user behaviors. Figure 4 shows an example of two PMFs of rating values, corresponding to two pair of users in different households. The one on the left emphasizes the fact that user u322924 (thick lined) rarely gives ratings higher than 90 points. On the contrary, user u880228 (dashed lined) usually gives ratings higher than 90 points. The example on the right has a stronger discrimination. The dashed user rates with less than 10 points most of the time. On the contrary, the thick lined user tends to rate over 60 points.

The analysis presented above suggests us to take into account the following dimensions in order to identify raters:

- The **hour of the day** in which a user rates movies more frequently (H).
- The **day of the week** in which a user rates movies more frequently (W).
- The **date of rating** (D).
- The **number of ratings** given by a user (R).

These findings motivated us to use probability-based models in order to classify users in a household depending upon the described dimensions. Previously to their application, we studied possible classifiers to be used, as well as more traditional recommendation models which could serve as base-lines.

3. RELATED WORK

3.1 Classification Models

With respect to classification, there are many different paths to explore. Inductive learning can be defined as those methods trying to induce a general rule from a set of observed instances. Frequently used and well known are, among others, the information theoretic-based ones (e.g. Rocchio classifier [19]), decision trees (like the C4.5 algorithm [18], the evolved algorithm of the famous Quinlan’s ID3 [17]), instance-based methods (including Nearest Neighbor (kNN)-based models [5]), and probabilistic classifiers (e.g. Naïve Bayes or simple Bayesian classifier [6]). As we can observe, depending on the context under study it is possible to choose between many different models to classify items. One key

circumstance in the decision about which model to use is the existence or not of prior knowledge. Prior knowledge plays a major role. Usually the way to classify is highly dependent on the existence of examples (used as training data), and the features of the data under study. When prior knowledge of the patterns to be classified is available, existing methods differentiate between supervised and unsupervised learning. In the first case the training data are provided with the classes to which the examples belong to. In the other case no prior classification exists, and the user will set hypothesis about the number of classes to be generated (as for example by means of clustering techniques [9]).

When dealing with simpler Bayesian classifiers, independence is also an important aspect to be considered. When features are independent, binary models usually leads to simplified linear classifiers. The more independent we can make the classifiers, the simpler the classifier can be. Naïve classifiers are those under the assumption that the attributes are independent given the class. In [6], it is shown that these simple Bayesian classifiers can be optimal under zero-one loss (i.e. the misclassification rate). There exist innovative proposals as the one described in [13], to extend the modeling flexibility of Naïve Bayes models by introducing latent variables to relax some of the independence assumptions in these models.

In the case of the MoviePilot dataset, there is a vast quantity of training data, all of it with proper class assignments. As showed in the brief study made above, there is a lot of information about the a priori behavior of the users being part of each of the households, by means of the characteristics known about them, that is, their features. Features should be as much discriminant as possible. However, in this case, there are not many possibilities to extract features useful for the analysis from the data. Some of the ones analyzed are directly extracted from the data itself (as for example the hour of the day (H)), but others are derived ones (e.g. the date of rating (D)).

Because of their usefulness, many disciplines use classification methods based on supervised learning. For example, Pang et al. [15] employed three different machine learning methods (Naïve Bayes, maximum entropy classification and SVM) to classify documents by overall sentiment, using movie reviews as data. In another context, Herrera et al. [11] used both kNN and Naïve Bayes methods for in-

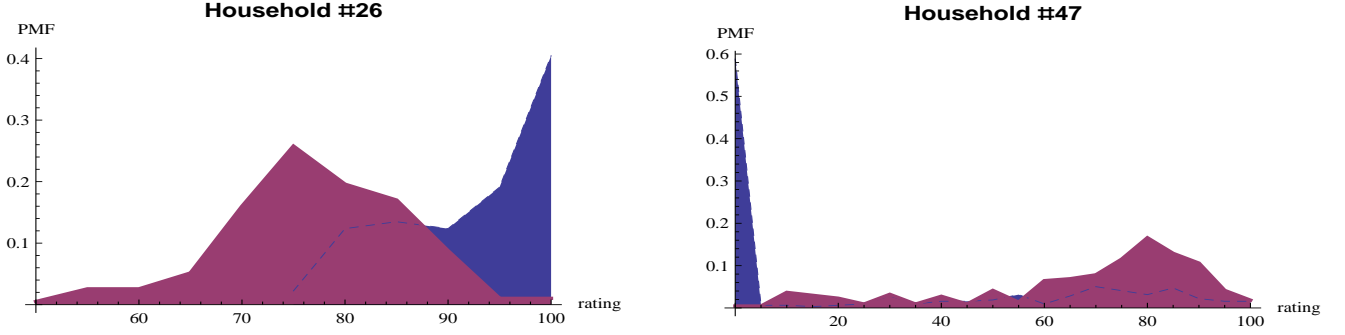


Figure 4: Histogram of the users rating values in households #26 (left) and #47 (right)

strument classification in music recognition. Additionally, in [8] authors studied the problem of personalized tag recommendation for Flickr using, among others, a Naïve Bayes approach.

As we can observe, there exist many applications on diverse contexts. The models used should be chosen adapted to the characteristics of the problem being solved. In what follows, the problem we focus on is to identify the ratings of movies (i.e., determining which user performed a rating), when different sized households (group of users) are considered.

3.2 Recommendation Models

Recommender Systems apply techniques from statistics and knowledge discovery to the problem of recommending items to users of a system [20]. The most common used approach is collaborative filtering (CF) [21], which tries to predict the utility of items for a target user based on items previously liked by the target and/or other users. The information about user interest is called the user profile. There are also content-based (CB) models [16], which search for items similar to other items that the user liked in the past, using descriptions of the items and the user profile. Thus, CB recommendation models require an explicit description of items. Due to limitations from both approaches, there are also hybrid approaches [4], which combine elements from both CF and CB. Due to the fact that the Moviepilot dataset do not contain descriptions about items, the only choice herein is CF.

CF recommendation algorithms can be further classified into heuristics and model-based ones [3]. The former make use of the entire collection of user profiles to compute predictions (e.g. in kNN-based recommendation models [10]), whilst the latter learn a model, which is then used to compute predictions (e.g. in Matrix Factorization-based recommendation models [12]).

4. PREDICTIVE MODELS

This section describes the models used for the challenge. We begin with the probabilistic models which turned out to give the best performance results. Then, we describe other more traditional (ad-hoc) recommendation models which were used to compare our results.

4.1 Probability-based Models

The findings observed from the dataset analysis motivated us to use probability-based models to classify which users were the ones who evaluated each movie as required by the

challenge. We used a discriminant function based on the PMFs obtained, giving more probability to users depending on the probabilities of the previously mentioned dimensions of information, namely time and number of ratings. We describe below our two approaches.

4.1.1 A priori Model

Let us consider a set of objects $O = \{o_1, o_2, \dots, o_m\}$ and a set of classes $\Omega = \{\omega_1, \omega_2, \dots, \omega_c\}$, such that each object o_i is member of one, and only one, class ω_j . In addition, consider that these objects are described by means of the value of some numerical quantity feature, called X . Now, the question we want to answer herein is whether it is possible to determine which class an object o_i belongs to or not, once the value x_i of its feature X is already known. If we assume that we know the a priori probabilities of the respective classes, a simple classification rule can be:

$$\text{Assign } o_i \text{ to } \omega_j^* = \arg \max_{\omega_j \in \Omega} P(X = x_i | \omega_j)$$

Bringing this model to our case, let U_h be the set of users from household h , and let $\tilde{R}_h = \{\tilde{r}_1, \tilde{r}_2, \dots, \tilde{r}_m\}$ be the set of unidentified ratings from h , that is, ratings that are known to be given by a user u_j from U_h , but not knowing which particular user u_j gave it. We define, based on the a priori PMFs of feature X , $P(X|u_j)$ (where X can be any of the information dimensions described in Section 2.2):

$$\text{score}(\tilde{r}_i, u_j) = P(X = x_i | u_j)$$

Once the scores given to each pair (\tilde{r}_i, u_j) are determined, the a priori-based discriminant function assigns the rating \tilde{r}_i to the user that reached the highest probability. That is:

$$\text{Assign } \tilde{r}_i \text{ to } u_j^* = \arg \max_{u_j \in U_h} P(X = x_i | u_j) \quad (1)$$

4.1.2 Naïve Bayes Model

Now, considering we know the PMFs of feature X and each class ω_j , i.e., $P(X)$ and $P(\omega_j)$, by means of applying the Bayes' theorem, we compute the corresponding probabilities of each class provided the feature X :

$$P(\omega_j | X = x_i) = \frac{P(X = x_i | \omega_j)P(\omega_j)}{P(X = x_i)}$$

Then, the previous classification rule is rewritten as:

$$\text{Assign } o_i \text{ to } \omega_j^* = \arg \max_{\omega_j \in \Omega} P(\omega_j | X = x_i)$$

Therefore, in our case we compute again the previously defined scores as:

$$\text{score}(\tilde{r}_i, u_j) = P(u_j | X = x_i)$$

Then, we can apply the same decision rule as defined in the previous model (1).

These models can be easily extended to consider a set of features $\mathcal{X} = \{X_1, X_2, \dots, X_n\}$ describing each object o_i by computing the combined probability $P(X_1 = x_{1_i}, X_2 = x_{2_i}, \dots, X_n = x_{n_i} | \omega_j)$. Using the conditional independence (a.k.a. naïve) assumption that each feature X_k is conditionally independent of every other feature X_l for $k \neq l$, we can compute the combined probability as $\prod_{k=1}^n P(X_k = x_{k_i} | \omega_j)$ [23].

4.2 Rating-based Models

A different discriminant can be build if, instead of inferring which user is more likely to rate a particular item, we compute a prediction of the rating that a user u_j would give towards an item m_i , that is, \hat{r}_{u_j, m_i} . Thus, if we compute all the rating predictions for m_i for each user in U_h , and knowing the actual rating value of \tilde{r}_i (included in the data as provided by the challenge), we can assign the rating to the user whose prediction is the closest, that is:

$$\text{Assign } \tilde{r}_i \text{ to } u_j^* = \arg \min_{u_j \in U_h} |\tilde{r}_i - \hat{r}_{u_j, m_i}|$$

Since this strategy depends on how predictions are made, we have used two state-of-the-art recommendation methods in order to compute these predictions: nearest neighbors (an example of heuristic CF) and matrix factorization (an example of model-based CF).

4.2.1 Heuristic CF

We used a kNN CF model [14], as a heuristic CF baseline. It is a widely used recommendation method due to its simplicity and good performance. It considers a set of most similar items (*nearest neighbors*) with respect to the target item. Then, it extrapolates the rating the target user would give to the item, by using the similarity value as a weighting factor and the ratings given to the neighbors by the target user in the following way:

$$\hat{r}_{u,i} = b \sum_{i' \in N_k(i)} \text{sim}(i, i') \times r_{u,i'}$$

Here b is a normalizing factor, usually computed as $b = 1 / \sum_{i' \in N_k(i)} \text{sim}(i, i')$, $\text{sim}(i, i')$ is the similarity value between items i and i' , usually computed as the correlation among co-ratings, or Pearson Correlation:

$$\text{sim}(i, i') = \frac{\sum_{u \in \mathcal{U}_{i, i'}} (r_{u,i} - \bar{r}_i)(r_{u,i'} - \bar{r}_{i'})}{\sqrt{\sum_{u \in \mathcal{U}_{i, i'}} (r_{u,i} - \bar{r}_i)^2 \sum_{u \in \mathcal{U}_{i, i'}} (r_{u,i'} - \bar{r}_{i'})^2}}$$

where $\mathcal{U}_{i, i'} = \{u \in \mathcal{U} : r_{u,i} \neq \emptyset \wedge r_{u,i'} \neq \emptyset\}$, that is, the set of users that rated both items. The set $N_k(i)$ represents the k nearest neighbors of i computed as (with $N_0 = \emptyset$):

$$N_k(i) = \bigcup_{j=1}^k i'_j : i'_j = \arg \max_{i' \in \mathcal{I} - N_{j-1}(i), i' \neq i} \text{sim}(i, i')$$

This method is known as a *item-based* CF, since it is based on rating information of similar items. In the same way, this model can be computed on users similar to the target user, in which case it is called *user-based* CF [10].

4.2.2 Model-based CF

In the case of model-based CF, we selected a Matrix Factorization (MF) model baseline. It is an adaptation of the Singular Value Decomposition approach that is gaining increasing interest in the field of Recommender Systems due to its good performance [12]. In this technique, the known rating values, represented as a rating matrix \mathcal{R} , are iteratively approximated by user and item factor matrices \mathcal{P} and \mathcal{Q} of lower dimension (f in our notation) such that:

$$\hat{r}_{u,i} = \sum_{j=0}^f \mathcal{P}_{u,j} \cdot \mathcal{Q}_{j,i} = \mathcal{P}_u^T \mathcal{Q}_i$$

One advantage of this approach is that \mathcal{P} and \mathcal{Q} values may be computed for all users and items using only the known values \mathcal{R} , minimizing an estimation of the difference, e.g. the Frobenius Norm: $\min \|\mathcal{R} - \mathcal{P}\mathcal{Q}\|^2$. Overfitting can be alleviated using regularization, i.e. penalizing the magnitude of the approximated vectors [12]. The common regularized formulation for collaborative filtering is inspired in minimizing the squared error on the set of ratings:

$$\min_{\mathcal{P}^*, \mathcal{Q}^*} \sum_{u, i \in \mathcal{R}} (r_{u,i} - \mathcal{P}_u^T \mathcal{Q}_i)^2 + \lambda (\|\mathcal{P}_u\|^2 + \|\mathcal{Q}_i\|^2)$$

Different algorithms exist to compute this kind of factorization. A widely used implementation of stochastic gradient descent was published by Simon Funk¹ in the context of the Netflix Prize. In this implementation, for each known rating, the parameters are optimized by updating them in the opposite direction of the gradient of the optimization criterion, using a *learning rate* parameter γ which controls the amount of update [12, 22]:

$$\begin{aligned} p'_u &\leftarrow p_u - \gamma \cdot \frac{\partial J}{\partial p_u} \\ q'_i &\leftarrow q_i - \gamma \cdot \frac{\partial J}{\partial q_i} \end{aligned}$$

5. EXPERIMENTS

5.1 Implementation details

Table 2 shows the parameter values used in the implementation of the rating-based models described in Section 4.2. Note that we used an item-based kNN algorithm. In the case of the probabilistic-based models, we ran out several trials combining the different features previously defined. In the next section, we show the best results obtained with all the described algorithms.

5.2 Evaluation Methodology

We have tested the accuracy of the models in terms of the classification error rate by household, i.e., the number of correct predictions divided by the number of predictions, averaged by household, in agreement with the rules of the challenge. Let HH denote the full set of households in the challenge data, and $f(\cdot)$ the model under evaluation. The classification error rate can be expressed by:

¹<http://sifter.org/~simon/journal/20061211.html>

Table 2: Parameter values

Model	Param.	Value
kNN	k	200 items
MF	f	10 factors
	λ	0.001
	γ	0.02

$$E_{HH} = \frac{1}{|HH|} \sum_{h=1}^{|HH|} \frac{1}{|HH_h|} \sum_{(o_i, \omega_i) \in HH_h} L(\omega_i, f(o_i))$$

where

$$L(\omega, \hat{\omega}) = \begin{cases} 1 & \text{if } \omega = \hat{\omega} \\ 0 & \text{otherwise} \end{cases}$$

Now, let HH^k denote the set of households of size k ($HH = \bigcup_k HH^k$). Then, the classification error rate per household size is:

$$E_{HH^k} = \frac{1}{|HH^k|} \sum_{h=1}^{|HH^k|} \frac{1}{|HH_h^k|} \sum_{(o_i, \omega_i) \in HH_h^k} L(\omega_i, f(o_i))$$

We computed an additional set of metrics based on precision, such as Precision at level 5 (P@5) and 10 (P@10), Mean Average Precision (MAP) and Area Under the Curve (AUC), computed on each user's recommendation list and averaged on all test users (not on a per-household basis). Note that a recommendation list in this context can be easily constructed for a target user by ordering the full set of items, putting in the top places those items believed to have been rated by the target user. That is, we sort the items classified as rated by the target user with respect to the score obtained by the probability-based models, or the predicted rating in the case of rating-based models, putting after them the remaining items in the dataset (except for those in the user profile) with a score/rating equal to 0.

Precision and MAP come from the Information Retrieval field [2], and are useful to measure how good a recommender is based on a given ranking. AUC is an additional metric commonly used in the Machine Learning community for measuring the classification error and how far a particular method is from a random classifier (which obtains an AUC of 0.5) [7]. For the four metrics included, the higher, the better, considering that their maximum value is 1.0.

5.3 Results

Table 3 shows results obtained with the tested models (bold indicates the best column value). Recall that R represents the number of ratings used as a feature for the probability-based models, H represents the hour of the day, W the day of the week, and D the date of rating; finally, a combination of those letters represents, obviously, a combination of those features.

It can be seen that the best performing algorithm is the a priori model when using the combination of **hour of the day** and **date of rate** features (HD). It is also interesting to note that, in general, a priori models have superior performance than Bayes models, independently of the features considered. A possible explanation for this is that the independence assumption is violated. Deeper analysis is required

Table 3: Results on Classification Error Rate (averaged by household size) for the Identifying Rating Task

Model	Household size			
	All	Size-2	Size-3	Size-4
A priori (R)	0.6180	0.6336	0.3998	0.3255
A priori (H)	0.9056	0.9074	0.8976	0.8065
A priori (W)	0.8683	0.8706	0.8257	0.8615
A priori (D)	0.8784	0.8800	0.8269	0.9527
A priori (RH)	0.9097	0.9115	0.9033	0.8060
A priori (RW)	0.8852	0.8877	0.8514	0.8383
A priori (RD)	0.8975	0.8991	0.8526	0.9456
A priori (HW)	0.9365	0.9350	0.9586	0.9567
A priori (HD)	0.9392	0.9375	0.9604	0.9803
A priori (DW)	0.8825	0.8837	0.8419	0.9487
A priori (HRDW)	0.9374	0.9358	0.9572	0.9773
Naïve Bayes (R)	0.6839	0.6979	0.5175	0.3145
Naïve Bayes (H)	0.9049	0.9084	0.9033	0.6688
Naïve Bayes (W)	0.8597	0.8654	0.7766	0.7673
Naïve Bayes (D)	0.8543	0.8627	0.7111	0.7915
Naïve Bayes (RH)	0.8858	0.8883	0.8890	0.7067
Naïve Bayes (RW)	0.8726	0.8767	0.8276	0.7537
Naïve Bayes (RD)	0.8579	0.8652	0.7093	0.8837
Naïve Bayes (HW)	0.9211	0.9214	0.9315	0.8675
Naïve Bayes (HD)	0.9140	0.9154	0.8908	0.8968
Naïve Bayes (DW)	0.8651	0.8707	0.7472	0.8994
Naïve Bayes (HRDW)	0.9191	0.9188	0.9192	0.9376
kNN	0.6467	0.6580	0.4865	0.4399
MF	0.6412	0.6525	0.5016	0.3668

in order to verify if the independence assumption between features is acceptable or not.

All the results involving the H feature, considered alone or combined with other features, present a value up to 0.9 except for the case of Bayes (RH) within all the households (second column in Table 3). No other algorithm outperforms this value. This fact gives us a strong evidence of the importance of this feature. Among the three time-aware features studied (H, D and W), H is the one with higher discriminant capabilities for the task analyzed in this work.

It is also remarkable the poor performance of the number of ratings feature (R). It gives the lowest values for the metric considered, even lower than the baselines, in this case, the rating-based models.

Regarding the classical recommendation models, which are based on the extrapolation of rating values, both of them present poorer results than most of the probabilistic ones. The only probabilistic models that are comparable to them are the ones based on the rating value feature. This seems to remark that discriminating user ratings based only on rating values is hard, and in fact, other features (such as the temporal ones) are better suited for this task.

Table 4 shows the best results using an additional set of metrics, based on precision such as P@5, P@10, and MAP, and AUC (area under the curve). As it may be seen, results are consistent with classification accuracy rate outcome, regarding the best performing models, and besides, the obtained results are very high for the proposed probabilistic models.

6. CONCLUSIONS AND FUTURE WORK

This paper has described methods able to identify users that made particular ratings. We focused the analysis on the study of probability mass functions of the available features describing ratings, thus developing well-performing probabi-

Table 4: Additional metrics for the task

Model	P@5	P@10	MAP	AUC
A priori (HD)	0.9392	0.9375	0.9604	0.9803
Naïve Bayes (HW)	0.9211	0.9214	0.9315	0.8675
kNN	0.6287	0.4541	0.5509	0.8217
MF	0.6098	0.4468	0.5482	0.8279

lity-based models. The results obtained, when compared with the performance of rating-based models adapted for the task, show that an adequate combination of features allows probability models to obtain an interesting classification accuracy rate ($> 90\%$). It is also notable the good performance of the feature **hour of the day** combined with **date of rating** or **day of the week**, showing that users have “temporal habits” when rating movies. It is, thus, expectable that the addition of time data awareness into rating-based models improve their results. Furthermore, this finding could help on other interesting recommendation-related tasks, e.g. detecting the best hour of the day to send recommendations to users (via mobile devices, for instance).

Regarding future work, we will test additional discriminant functions, based on clustering, SVMs, etc. Moreover, we think that the usage of classifiers specific for binary classes may improve performance on 2-sized households, whereas multi-class classifiers should be used on 3 and 4-sized households. Finally, a mixture of classifiers can be considered for further improvements on classification accuracy. We also contemplate to study the independence assumption of the considered features using, for example, Fisher’s independence analysis based on contingency tables.

7. ACKNOWLEDGMENTS

This work is supported by the Spanish Government (TIN 2008-06566-C04-02) and by the Comunidad de Madrid and Universidad Autónoma de Madrid (CCG10-UAM/TIC-5877). The authors acknowledge support from CCC at UAM.

8. REFERENCES

- [1] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145, 2005.
- [2] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc, Boston, MA, USA, 1999.
- [3] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. pages 43–52. Morgan Kaufmann, 1998.
- [4] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.
- [5] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.
- [6] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Mach. Learn.*, 29(2-3):103–130, 1997.
- [7] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2000.
- [8] N. Garg and I. Weber. Personalized, interactive tag recommendation for flickr. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys ’08, pages 67–74, New York, NY, USA, 2008. ACM.
- [9] J. A. Hartigan. *Clustering Algorithms (Probability & Mathematical Statistics)*. John Wiley & Sons Inc, 1975.
- [10] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR ’99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, New York, NY, USA, 1999. ACM.
- [11] P. Herrera, X. Amatriain, E. Batlle, and X. Serra. Towards instrument segmentation for music content description: a critical review of instrument classification techniques. In *Proceedings of the 1st International Symposium on Music Information Retrieval*, 2000.
- [12] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- [13] H. Langseth and T. D. Nielsen. Classification using hierarchical naïve bayes models. *Mach. Learn.*, 63(2):135–159, 2006.
- [14] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.
- [15] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP ’02, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.
- [16] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The Adaptive Web: Methods and Strategies of Web Personalization.*, volume 4321, 2007.
- [17] J. R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, 1986.
- [18] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 1993.
- [19] J. Rocchio. *Relevance Feedback in Information Retrieval*, pages 313–323. The SMART Retrieval System. 1971.
- [20] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM conference on Electronic commerce*, EC ’00, pages 158–167, New York, NY, USA, 2000. ACM.
- [21] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009, 2009.
- [22] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Investigation of various matrix factorization methods for large recommender systems. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition - NETFLIX ’08*, pages 1–8, 2008.
- [23] I. H. Witten, E. Frank, and M. V. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2nd edition, 2005.