Empirical Analysis of Predictive Algorithms for Recommender Systems

Ruoheng Li RUOHENG@EDEN.RUTGERS.EDU
Statistics Department, Rutgers University, NJ 08854 USA

Abstract

Recommeder systems improve access to relevant products and information by making personalized suggestions based on previous examples of a user's likes and dislikes. Most existing recommender systems use collaborative filtering methods that base recommendations on other users' preferences. In this paper we mainly describe two algorithms, designed for this task, including correlation coefficients and statistical Bayesian methods. In order to combine more available information into the system, we incorporate user ratings and user features into a single Bayesian model. In principle, it can address the "cold-start" issue. Finally, we experimentally compare the predictive accuracy of the various methods.

1. Introduction

Recommender systems have emerged as an important application area and have been the focus of considerable recent academic and commercial interest. It applies data analysis techniques to the problem of helping users find the items they would like to purchase by producing a predicted likeliness score or a list of top-N recommended items for a given user. Recommendations can be based on demographics of the users, overall top selling items or past buying habits of users as a predictor of future purchases.

Most existing recommendation systems almost exclusively utilize a form of computized mathchmaking called *collaborative filtering* or *social filtering* (Breese, Heckerman & Kadie 1998). The system maintains a database of the preferences for items by individual users. When a new user comes, he/she is matched against the database to discover other users whose known preferences are correlated significantly with this new user. Items that are recommended to him/her are those enjoyed by their matched users. Such approach is built on the assumption that a good way to find interesting content is to find other people who have similar tastes, and then recommend items that those users like.

In this paper, we describe various collaborative filtering prediction methods, such as the correlation coefficients, and algorithms based on the Bayesian model. Since collaborative filtering requires that a new user rate some items, so to provide a solution to such a "cold start" problem, we incorporate user ratings and user features into a single Bayesian model. We present empirical data on the predictive performance of the various algorithms.

2. Collaborative Filtering Algorithms

The task in collaborative filtering is to predict the utility of a certain item for a particular user based on the user's previous preferences and the opinions of other like-minded users.

Table 1 shows the data typically available to a recommender system. Here, the rows represent *n* users and columns represent *m* items. The entire $n \times m$ user-item data is considered as a rating matrix *V* in a collaborative filtering system. Each entry in *V* represents the preference vote (ratings) of the *i*th user on the *j*th item. The votes are binary, indicating whether or not each user likes each item. These elements could also take on a range of values that indicate the extent to which a user likes an item. Missing data in this matrix means unrated items. Each user also is associated with a vector of covariates (i.e. attributes of the users).

Table 1. Collaborative filtering data.

	I_1	I_2	I_3	…	I_M	
U_1	1	1	0	…	0	Past User
U_2	0	1	1	…	0	
U_3	0	1	1	…	0	
. . .						
U_N	0	0	1	…	0	
U_{NEW}	0	1	?	…	?	New User

The bottom row of Table 1 is the new user for whom the task of a collaborative filtering algorithm is to predict an item similarity. This predicted value is within the same scale as the opinion values provided by the past users.

Researchers have devised a number of collaborative filtering algorithms that can be divided into two main categories, *Memory-based* and *Model-based* algorithms (Breese, Heckerman, & Kadie 1998). Memory-based algorithms operate over the entire user database to generate a prediction. Model-based collaborative filtering, in contrast, uses the user database to estimate or learn a model, which is used for predictions.

2.1 Memory-Based Algorithms

Memory-based algorithms utilize the entire user database to make predictions. The predicted votes of the new user are based on some partial information regarding the active user and a set of weights calculated from the user database. We assume that the prediction of the new user for item j, $p_{a\,j}$, is a weighted sum of the votes of the other users:

$$p_{a,j} = \bar{v}_a + \kappa \sum_{i}^{n} w(a,i)(v_{i,j} - \bar{v}_i)$$

where n is the number of users in the collaborative filtering database with nonzero weights. The weights $w(a,i)$ can reflect distance, correlation, or similarity between each user i and the new user. κ is a normalizing factor such that the absolute values of the weights sum to unity.

In this paper, we analyze memory-based algorithms using Pearson correlation coefficients as the basis of the weights. The basic idea of the correlation method is to first compute the similarity between a new user and each of the past users in the database using:

$$w(u_{new}, u_i) = \frac{\sum_j (v_{new,j} - \bar{v}_{new})(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{new,j} - \bar{v}_{new})^2 (v_{i,j} - \bar{v}_i)^2}}$$

The summations are over only the items which both the new user and user i have rated.

The prediction for item j for the new user is then a weighted sum of the past users' votes for that item:

$$p_{new,j} = \bar{v}_{new} + \frac{\sum_i w(u_{new}, u_i)(v_{i,j} - \bar{v}_i)}{\sum_i w(u_{new}, u_i)}$$

where the summations are over those past users who have rated item j. Note we cannot user this method until a user have rated at least two items. This is the so-called "cold start" problem.

2.2 Model-based Algorithm

Model-based collaborative filtering algorithms provide item recommendation by first developing a model of user ratings. Algorithms in this category take a probabilistic approach and consider the collaborative filtering process as computing the expected value of a user prediction, given his/her ratings on other items. If we assume that the votes are integer with a range from 0 to m we have:

$$p_{a,j} = E(v_{a,j}) = \sum_{i=0}^{m} \Pr(v_{a,j} = i \mid v_{a,k} \in I_a) i$$

where the probability expression is the probability that the new user a will have a particular vote value of item j given the previously observed votes. The model building process is performed by different machine learning algorithms such as *Bayesian network*, *clustering*, and *rule-based* approaches.

In this paper, we examine the clustering model based on a Bayesian classifier where the probability of votes are conditional variable C taking on some relatively small number of discrete values. The idea is that there are certain groups or types of users capturing a common set of preferences and tastes. Given the class, the preferences regarding the various items (expressed by the votes) are independent. The probability model relating the joint probability of class and votes to a tractable set of conditional and marginal distributions is the standard "naïve" Bayes formulation:

$$\Pr(C = c, v_1, \cdots, v_n) = \Pr(C = c) \prod_{i=1}^{n} \Pr(v_i \mid C = c)$$

The left-hand side of this expression is the probability of observing an individual of a particular class and a complete set of vote values. It is straightforward to calculate the needed probability expression for the new user provided before.

The parameter of the model, the probabilities of class membership *Pr(C=c)*, and the conditional probabilities of votes given class *Pr(*v_i*|C=c)* are estimated from a training set of user votes, the user database. Since we have assigned a user covariate for each user in the database, the class variable is computed based on this vector. These prior distributions based on the training set lead directly to predictions. This can, in principle, provide an elegant solution to the "cold-start" problem, since initial recommendations use only the user covariates. Unlike the correlation coefficients method, the Bayesian model doesn't require the new user to first rate any item in order to compute the predictions. As data (i.e. user ratings) accumulate, these prior distributions are updated via Bayes rule.

3. Empirical Anaylsis

3.1 Datasets

Our data set is a small example in which we recommend beverages to the user. We gathered data from 210 "users" who rated how well they liked 16 different drinks on a

Table 2. Comparison of Brier Scores.

	Numbers of Drinks User has Rated				
Method	0	2	5	10	15
Past User's Mean	0.37125				
Past Users' Group Mean	0.34635				
Correlation	NA	0.34821	0.35714	0.32725	0.28217
Bayesian Model	0.36875	0.33376	0.31297	0.30181	0.29164

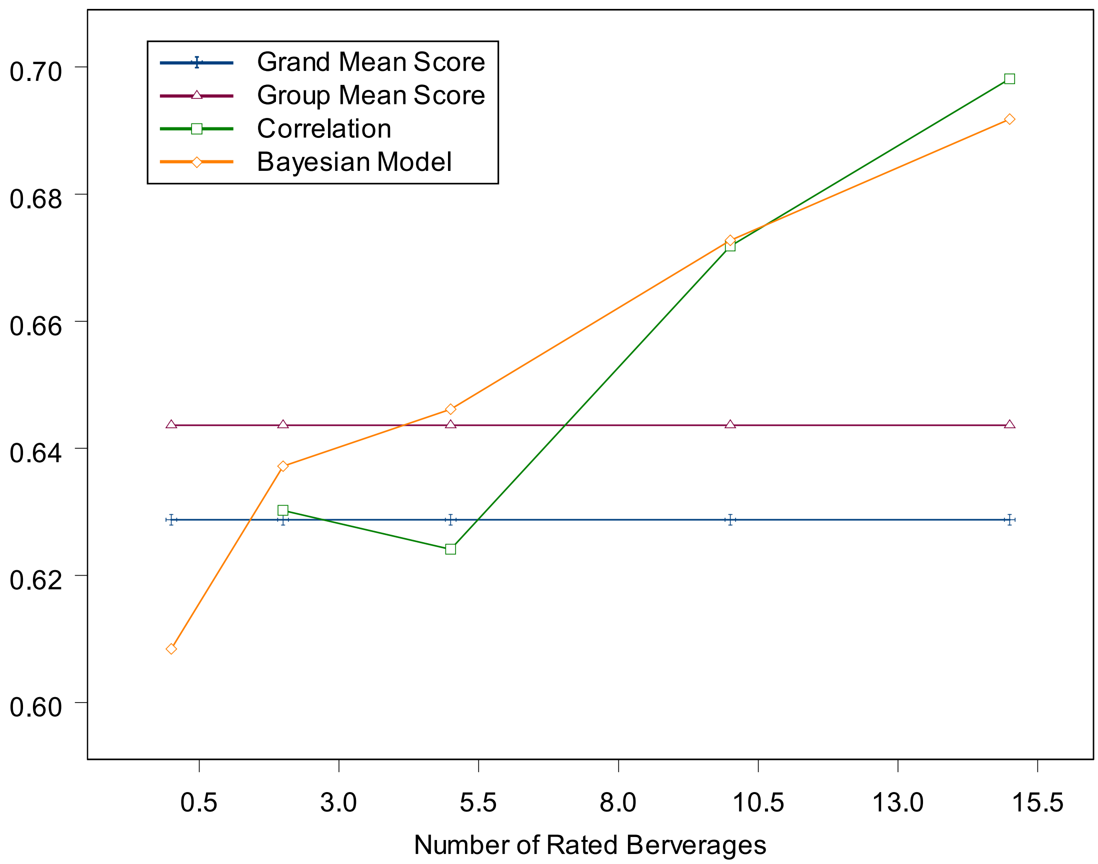

Figure 1. Comparison of classification accuracy.

scale of 1 to 7. The users were students in one statistics course at the University of Washington. We also collected data on each student, providing us with the following user covariates: age, gender and race. The average age of the users was 21.5, and 62% were female. The group was predominantly Caucasian (62%) and Asia (30%).

Before applying the collaborative filtering,, We dichotomized the votes. We decided that if the user gave the drink a rating greater than 4 then the user "likes" the drink, otherwise the user does not.

3.2 Evaluation Criteria

The effectiveness of a collaborative filtering algorithm depends on the manner in which recommendations will be presented to the user. To evaluate these algorithms, we have defined the metrics on the type of collaborative filtering application and interface one is providing.

For each of the new users, we compute the probability that the user will like each drink. We evaluate each method by comparing these probabilities with the users' true ratings. We applied two scoring metrics in our evaluations. First, we look at classification accuracy. If the probability for an item is greater than 0.5, we recommend that item to the user. Classification accuracy is the percentage of items correctly recommended or not recommended. Second, we look at the mean Brier Score (Brier, 1950) for each method. The Brier score is essentially the mean squared error for binary predictions, so lower scores indicate higher effectiveness.

In our case, we obtain a Brier score for each new user:

$$B_{new} = \frac{1}{m}\sum_{j=1}^{m}\left(v_{new,j} - p_{new,j}\right)^2$$

where $p_{new\ j}$ is the new user's prediction for item j and $v_{new\ j}$ is the new user's actual vote on that item. We then compute the average Brier score across all new users. The Brier score combines both accuracy and calibration in a single scoring rule.

3.3 Experiment Results

Beside the correlation coefficients and Bayesian model, we also examined two different approaches. First, for every user, we predict the mean score that past users gave on this item (the "grand mean"). Second, we predict the mean score that past user with the same user covariates gave the item (the "group mean").

We randomly select 160 users for the training set and keep the remaining 50 users as "new users" for testing. In order to reflect the different numbers of votes available to the recommender, we also examined the performance of the method after learning the ratings of the new users on some randomly selected items. For our dataset, we looked at the results on the remaining unrated beverages when the users rated 2, 5, 10 and 15 drinks. The experiment measures the algorithms' performance when given as much data as possible from each test user when we know the ratings of 15 drinks for the testing set. And the various experiments on 2, 5 and 10 drinks examine the performance of the algorithms when there is relatively little known about a new user.

Figure 1 and Table 2 show the results. The "Group Mean score" gave better results than the "Grand Mean score" since it included the user covariates data into the prediction. The Bayesian method performs well in general. It outperforms the correlation method, especially for the limited-data scenarios.

4. Conclusion

This paper presents an empirical experiment regarding the predictive performance of statistical algorithms for collaborative filtering in recommender systems. Results indicate for limited-data conditions, the Bayesian model performs well in general, outperforming the correlation method, since it can use partial information effectively. Because the user doesn't have to first rate any item to build the Bayesian model, it also address the "cold-start" issue. However results indicate that if given larger dataset the correlation method is very likely to be competitive with the Bayesian model. So between correlation and Bayesian model, the preferred method still depends on the size of the dataset and the availability of votes with which to make predictions.

References

Brier, G. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Reviews,* 75:1-3, 1950.

Sarwar, B., Karypis, G., Konstan, J., & Ried, J. (2001). Item-based Collaborative Filtering Recommendation Algorithms. Group/Army HPC Research Center Department of Computer Science and Engineering, University of Minnesota.

Condliff, M., Lewis, D., D., Madigan, D., & Posse, C. (1999), Bayesian mixed-effect models for recommender systems. *In ACM S[GIR '99 Workshop on Recommender Systems: Algorithms and Evaluation*.

Breese, J., Heckerman, D., & Kadie, C. (1998). *Empirical analysis of predictive performance for collaborative filtering* (Technical Report 1998). Microsoft Reserach.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, D.,Netes, D.,& Sartin, M. (1999). Combining Content-Based and Collaborative Filters in an Online Newspaper. *In ACM S[GIR '99 Workshop on Recommender Systems: Algorithms and Evaluation*.

Mooney, R., & Roy, L. (1999). Content-Based Book Recommending Using Learning for Text Categorization. *In ACM S[GIR '99 Workshop on Recommender Systems: Algorithms and Evaluation*.