Evaluating Information Filtering Techniques in an Adaptive Recommender System

John O'Donovan and John Dunnion

Department of Computer Science,
University College Dublin,
Dublin, Ireland.
{John.ODonovan,John.Dunnion}@ucd.ie

Abstract. With the huge increase in the volume of information available in digital form and the increasing diversity of Web applications, the need for efficient, reliable, information filtering is critical. New algorithms that filter information for a specific taste are being developed to tackle the problem of information overload. This paper proposes that there is a substantial relative difference in the performances of various filtering algorithms as they are applied to different datasets, and that these performance differences can be leveraged to form the basis of an Adaptive Information Filtering System. We classify five different datasets based on a number of metrics, including sparsity, ratings distribution and user-item ratio, and develop a regression function over these metrics to predict the suitability of a particular recommendation algorithm for a previously unseen dataset. Our results show that the predicted best algorithm does perform best on the new dataset.

1 Introduction

In order to provide a more personalised and tailored service to their users, an increasing number of on-line services provide recommendations to their users, eg eBay, Amazon.com, etc. With the huge increase in the volume of information available in digital form and the increasing diversity of Web applications, the need for efficient, reliable, information filtering is critical. New algorithms that filter information for a specific taste are being developed to tackle the problem of information overload. There are two main techniques upon which these algorithms are based: Collaborative Filtering [4][13] and Content-Based Filtering [7]. In this paper, we focus on the collaborative approach, where users provide ratings for items in a particular domain, and the system exploits similarities and differences between users based on item ratings to compute its recommendations.

Collaborative Filtering (CF) is a broad term for the process of recommending items to users based on similarities in taste. The underlying principle of Collaborative Filtering is the following: if users A and B rate k items similarly, they are considered to share similar tastes and should rate other items similarly. Collaborative Filtering techniques have the following advantages: they do not require items to be machine-analysable (as explained in [7]); they can arrive at serendipitous recommendations, that is, they can recommend relevant items that are completely different from those in a user's profile; and they require little knowledge engineering overhead. However, CF techniques are also subject to two serious restrictions. The first of these is the Sparsity Problem. In any given case, it is unlikely that two users have co-rated many of the items in the system. Accurate similarity measurements depend on rich user profiles with high overlap, which can be costly to attain. The second restriction is the Latency Problem, which affects new, unique or esoteric items. Such items will not be recommended by a system until they are included in a sufficient number of user profiles, as outlined in [14].

Similarity can be computed for CF by several well-known techniques, such as Cosine Similarity, Spearman's Correlation and Pearson's Correlation [9]. For our similarity calculations we employ Pearsons Correlation, as it is the most widely used and allows for better comparison with other systems. It may be defined as follows:

$$corr_{x,y} = \frac{\sum_{u \in U}(R_{u,x} - \overline{R}_x)(R_{u,y} - \overline{R}_y)}{\sqrt{\sum_{u \in U}(R_{u,x} - \overline{R}_x)^2 \cdot \sum_{u \in U}(R_{u,y} - \overline{R}_y)^2}}$$

where $corr_{x,y}$ is the Pearson correlation coefficient between users x and y, $R_{i,j}$ is the rating of item i by user j, and $\overline{R}_i$ is the average item rating by user i.

Different approaches to collaborative filtering and different implementations will be affected to varying degrees by the problems mentioned above. Performance will also be dependent on the dataset to which collaborative filtering is applied and the information used to compile a similarity model. For example, in a situation where the set of items to be recommended is relatively small and static and there is a large number of users, it would be better to employ an item-based approach [12][6] to collaborative filtering, since the similarity model is built up over the large number of user profiles. In such a situation, a user-based filtering approach (as in [1]) would not perform as well since there would be insufficient items in each profile to provide the level of overlap required for a reliable similarity model. It is this type of observation that we endeavour to capture and exploit in this work.

2 Adaptive Information Filtering

In this paper we introduce the AdRec system, an adaptive recommender which attempts to overcome the inherent difficulties with individual filtering techniques by employing an adaptive approach to the design of the filtering engine [15]. To achieve this adaptability in our system, we make the assumption that our datasets can be adequately described (for CF purposes) by a set of their salient features, which we use for classification. These features include user-item ratio, sparsity, density distribution and data type. We tested three collaborative recommendation algorithms (User-Based CF, Item-Based CF and Rule-Based CF) on four different experimental datasets (EachMovie, PTV, Jester and MovieLens), and noted the relative performance of each method with respect to these classification metrics. Using this information it was possible to develop a regression function for algorithm prediction based on these metrics alone. We tested the performance of this function by introducing another dataset, SmartRadio [5]. This set was classified according to the metrics, and the regression function was applied to the resulting values to attain an algorithm prediction.

If our system is successful and we can successfully perform this algorithm prediction task, it can form the basis of a generic recommender system, which can employ cutting-edge filtering techniques to a given system without having to manually tailor the recommendation engine for that system. The design of the system is completely modular, allowing new techniques to be added as they are developed.

The architecture of the AdRec system is presented in Figure 1.

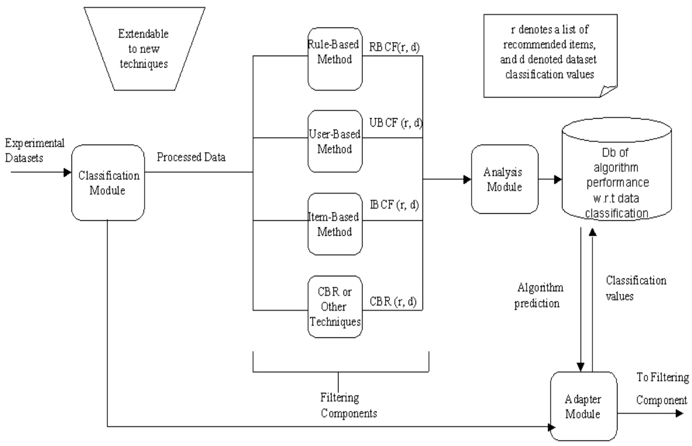

Fig. 1. AdRec System Architecture.

2.1 Regression

A linear regression model [12] is built up using our evaluations in [8]. This is based on a predictive function of several variables, as described in [10]:

$$E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

Values for each β_i are obtained by taking the classification metric values for each dataset, together with the best performing algorithm for each set, and solving the resulting system of simultaneous equations. A new, unseen dataset is classified in the same way and metrics produced. The regression function is then applied to this information to predict the algorithm giving the best performance on the new dataset.

3 Experimental Evaluation

The goal of the system is to predict the best-performing algorithm using only the regression function learned from the classification metrics of the other datasets, and the values the new dataset has for these metrics. Having calculated the regression function from tests shown in [8], we run all of the algorithms again on the new dataset. We have evaluated the system using predictive accuracy (mean absolute error and ROC Sensitivity [7]). However, here we present only a basic predictive accuracy test. These tests are simplified by keeping the neighbourhood size k and the test-train ratio constant at 30 and 80, respectively. These are optimal values we discovered empirically and reported in [8].

3.1 Experimental Data and Dataset Classification

For the initial training phase of the system, four experimental datasets were used: Jester [3] (an experimental dataset of jokes ratings, consisting of 21800 users ratings of 100 jokes); EachMovie (73000 user ratings of 1628 movies); PTV [2] (622 user ratings of TV programmes); and MovieLens [11] (100000 user ratings in the movie domain). In future work, it is hoped to also include a customer-product purchase database from an on-line sales company. For the purposes of our testing we selected subsets of 900 profiles from each of the above datasets comprised of the largest profiles, ie those users who had rated 20 items or more (with the exception of PTV, which only contains 622 profiles, and SmartRadio, which has 395). For processing by our system, the datasets were parsed, converted into the same format and stored in an SQL database. The SmartRadio dataset was classified according to the same metrics as the others and was found to be over 99% sparse and to have a user-item ratio of 1:9 ($\beta_1 = 0.99$ and $\beta_2 = 0.111$). A summary of the dataset classification is presented in Table 1.

Dataset	*User:Item*	*Sparsity (%)*	*Type*
PTV	1:6	94.25	TV Programmes
MovieLens	9:13	63.86	Movie Ratings
Jester	9:1	54	Jokes Ratings
EachMovie	9:17	33.97	Movie Ratings
SmartRadio	1:9 (approx)	99.98	Music Ratings

Table 1. Classification of Experimental Datasets.

3.2 Experimental Procedure

In this paper, we use predictive accuracy as the performance metric for the recommendation algorithms. For each dataset, if a user's rating is above a certain threshold, the item is considered "liked" by that user. This level of granularity was chosen because users will use rating scales differently. We tailored the threshold value for each individual scale, based on distribution of ratings. We predict the "liked" items for the unseen test data and record accuracy for each dataset. User profiles are split into training and test data. The training data is fed to each filtering component, which generates its own predictions for the unseen test data. To build our regression model, we use our results from [8].

3.3 Experimental Results

To validate our system, we need to show that the algorithm predicted by the regression function performs better than its competitors on the new dataset. The graph in Figure 2 shows that the user-based CF algorithm has a better predictive accuracy than its competitors. This algorithm was correctly predicted as the best performer by our regression function.

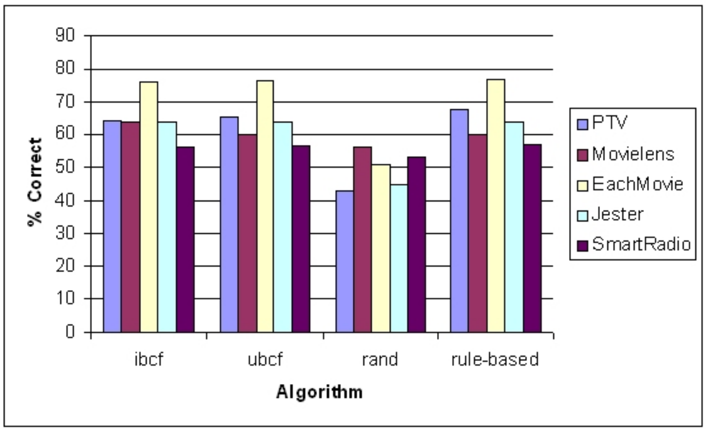

Fig. 2. Recommendation Accuracy for each algorithm, keeping k constant at 30 and the test-train ratio at 80%.

4 Conclusions and Future Work

In this paper we have presented the AdRec system. The approach adopted is based on a predictor for filtering techniques. More than merely developing specific filtering implementations, we have developed an information filtering architecture, capable of incorporating new technologies as they are developed. One application of this adaptive recommender could be commercially deployed in situations where system developers do not have the time or expertise available to assess which information filtering technique best suits the individual requirements of their application.

The testing procedure presented here will be enhanced to incorporate tenfold cross-validation and decision support metrics, such as Receiver Operator Characteristic (ROC), and better statistical accuracy in the form of mean absolute predictive error. Future work will also include applying the system to test data from other domains. This will provide a more reliable test-bench and hence a better regression function upon which future algorithm predictions may be based.

Acknowledgements

The support of the Enterprise Ireland Informatics Research Initiative is gratefully acknowledged. The research was funded under grant PRP/00/INF/06.

References

1. Daniel Billsus and Michael J. Pazzani. Learning collaborative information filters. In *Proc. 15th International Conf. on Machine Learning*, pages 46–54. Morgan Kaufmann, San Francisco, CA, 1998.
2. Paul Cotter and Barry Smyth. PTV: Intelligent personalised TV guides. In *Proceedings of the 7th Conference on Artificial Intelligence (AAAI-00) and of the 12th Conference on Innovative Applications of Artificial Intelligence (IAAI-00)*, pages 957–964, Menlo Park, CA, July 30– 3 2000. AAAI Press.
3. Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.
4. Nathaniel Good, J. Ben Schafer, Joseph A. Konstan, Al Borchers, Badrul Sarwar, Jon Herlocker, and John Riedl. Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI-99); Proceedings of the 11th Conference on Innovative Applications of Artificial Intelligence*, pages 439–446, Menlo Park, Cal., July 18–22 1999. AAAI/MIT Press.
5. C. Hayes, P. Cunningham, P. Clerkin, and M. Grimaldi. Programme-driven music radio, 2002.
6. George Karypis. Evaluation of item-based top-n recommendation algorithms. In *CIKM*, pages 247–254, 2001.
7. P. Melville, R. Mooney, and R. Nagarajan. Content-boosted collaborative filtering, 2001.
8. John O'Donovan and John Dunnion. A comparison of collaborative recommendation algorithms over diverse data. In *Proceedings of the National Conference on Artificial Intelligence and Cognitive Science (AICS), Ireland*, pages 101–104, September 17– September 19 2003.
9. Michael P. O'Mahony, Neil Hurley, and Guenole C. M. Silvestre. An attack on collaborative filtering. In *Proceedings of the 13th International Conference on Database and Expert Systems Applications*, pages 494–503. Springer-Verlag, 2002.
10. Adrian E. Raftery, David Madigan, and Jennifer A. Hoeting. Bayesian model averaging for linear regression models. *Journal of the American Statistical Association*, 92(437):179–191, 1997.
11. Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM CSCW'94 Conference on Computer-Supported Cooperative Work*, Sharing Information and Creating Meaning, pages 175–186, 1994.
12. Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John Reidl. Item-based collaborative filtering recommendation algorithms. In *World Wide Web*, pages 285–295, 2001.

13. Badrul M. Sarwar, Joseph A. Konstan, Al Borchers, Jon Herlocker, Brad Miller, and John Riedl. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In *Proceedings of ACM CSCW'98 Conference on Computer-Supported Cooperative Work*, Social Filtering, Social Influences, pages 345–354, 1998.
14. B. Smyth, D. Wilson, and D. O'Sullivan. Improving the quality of the personalised electronic programme guide. In *In Proceedings of the TV'02 the 2nd Workshop on Personalisation in Future TV, May 2002.*, pages 42–55, 2002.
15. M. Sollenborn and P. Funk. Category-based filtering in recommender systems for improved performance in dynamic domains. In *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Malaga, Spain. May 2002*, pages 436–439, 2002.