A Framework for Evaluation of Information Filtering Techniques in an Adaptive Recommender System

John O'Donovan and John Dunnion

Department of Computer Science,
University College Dublin,
Ireland.
{john.odonovan,john.dunnion}@ucd.ie

Abstract. With the current diversity of web-applications, the need for efficient, reliable, information filtering is more apparent now than ever before. New algorithms that filter information for a specific taste are being developed to tackle the information-overload problem. This paper proposes that there is a substantial relative difference in filtering algorithm performances 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 metrics such as sparsity, ratings distribution, user-item ratio etc, and develop a regression function over these metrics in order to predict suitability of a particular recommendation algorithm to a previously unseen dataset, using only the aforementioned metrics. Our provisional results show that the predicted best algorithm does perform better for the new dataset.

1 Introduction

Collaborative Filtering is a broad term for the process of recommending items to users based on similarities in taste. [4][13] An increasing number of online stores provide collaborative recommender systems on their sites, e.g. E-Bay, Amazon.com etc. There are two main bases upon which these systems operate: Collaborative and Content-Based Filters[7]. In this paper, we focus on the Collaborative approach, wherein users provide ratings for items in a particular domain, and the system exploits similarities and differences between users based on these to compute it's recommendations: If users A and B rate k items similarly, they share similar tastes and should rate other items similarly. There are many implementations and variants of these techniques in today's recommender systems. Collaborative Filtering (CF) techniques tend to 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 users profile. They also require little knowledge-engineering overhead. CF techniques are also subject to two serious restrictions: 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. These can be costly to attain. Latency Problem: This affects new, unique or esoteric items. These 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, auch as Cosine Similarity, Spearman's or Pearson's Correlation[9]. For all of our similarity calculations we employ Pearsons, as it is the most widely used and allows for better comparison with other systems.

$$corr_{x,y} = \frac{\sum u \in U(R_{u,x} - \overline{R}_x)(R_{u,y} - \overline{R}_y)}{\sqrt{\sum u \in U(R_{u,x} - \overline{R}_x)^2 \cdot \sum u \in U(R_{u,y} - \overline{R}_y)^2}} \tag{1}$$

$corr_{x,y}$ is the Pearson correlation coefficient between user x and user y. $R_{k,x}$ is the rating of user x on item k, and R_x is the average item rating by user x.

Different implementations and approaches to collaborative filtering will be affected to varying degrees by the problems mentioned above. Their performance will change based on the dataset that they operate on, and the information they harness to compile a similarity model. For example, for a situation where the set of items to be recommended is relatively small and static, and there are a large number of users, it would be advisable 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 this case 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.

2 Adaptive Information Filtering

In this paper we introduce the AdRec system, an adaptive recommender which attempts to overcome these inherent difficulties with individual filtering techniques by employing an adaptive approach to the filtering engine design[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 then tested our 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 note the relative performance differences of each method with respect to the classification metrics above. From this information it was possible to develop a regression function for algorithm prediction based on these metrics alone. We test the performance of this function by introducing another dataset. (SmartRadio [5]). This set is classified according to the required metrics, and the resulting values are run through the regression function to attain an algorithm prediction. If we can successfully perform this algorithm prediction task, we 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 systems design is completely modular, which has the advantage of allowing new techniques to be added as they develop.

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].

$$E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \tag{2}$$

Values for the $\beta_i's$ are gotten by going over all the classification metric values for each dataset, and the best performing algorithm for that set, and then solving a resulting system of simultaneous equations. The SmartRadio dataset was classified according to the same metrics as the others and was found to be over 99% sparse and have user-item ratio of 1:9. This information was put through the regression function, which predicted the user-based algorithm for best performance.

3 Experimental Evaluation

The aim 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 our regression function from tests shown in [8], we run all of the algorithms again on the new dataset. Predictive accuracy (mean absolute error and ROC Sensitivity[7]) are our main tests for this system, For this short paper however, we show 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 found in [8].

3.1 Experimental Data and Dataset Classification

For the initial training phase of the system, four experimental datasets are used: Jester[3] (an experimental dataset of jokes ratings, consisting of 21,800 users ratings of 100 jokes), EachMovie (73,000 users ratings of 1628 movies), PTV[2] (622 user ratings on TV programmes), and MovieLens[11] (100,000 user ratings in the movie domain.) It is hoped to also include a customer-product purchase database from an online sales company in the near future. For the purposes of our testing we selected subsets of 900 profiles from each of the above datasets comprised of the largest profiles: 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 modularity in our system, these datasets were all parsed into the same format and stored in an SQL database.

Table 1. Classification of Experimental Data

Dataset	Users-Items	Sparsity	Type
PTV	1:6	94.25%	TV Progs
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

3.2 Experimental Procedure

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

To validate our earlier proposal, we need to show that the algorithm predicted by the regression function performs better than its competitors on our new dataset.

3.3 Experimental Results

The graph below clearly shows that the user-based CF algorithm (predicted as the best-performer by our regression function) does in fact have a better predictive accuracy than all of its competitors.

4 Conclusion and Future Work

In this short paper we have forwarded the AdRec system. The approach adopted by this system is based on a predictor for filtering techniques. More than simply developing specific filtering implementations, we produce an information filtering architecture, capable of incorporating new technologies as they arrive. One application of this adaptive recommender could be commercially deployed in cases 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 in this paper will be reviewed in a further paper to incorporate 10-fold cross validation and decision support metrics such as Receiver Operator Characteristic (ROC) and better statistical accuracy in the form of

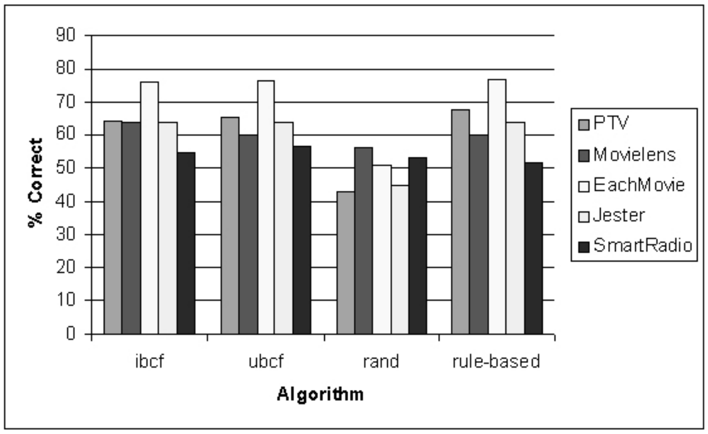

Fig. 1. Recommendation Accuracy for each algorithm. (Keeping k constant at 30 and test-train ratio at 80%)

mean absolute predictive error. Future work will also include extension of the scope of test data to other domains. These improvements should provide a more reliable test-bench and therefore a better regression function upon which to base our algorithm predictions.

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.