A Meta-Learning Approach To Recommendation Engine Design

John O'Donovan

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

Abstract. The demand for efficient and reliable information filtering techniques for digital data is greater than ever before. We present an assessment of the performance of three popular recommendation algorithms over a range of diverse data. Our aim is to show that the relative performance of these algorithms varies as they are applied to different data, and that these differences can be harnessed to develop a system which can predict the best performing algorithm for a dataset. [?][?].
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.

1 The AdRec System

AdRec [?] is a framework wherein recommendation techniques can be dynamically assigned to datasets based on a lightweight classification of that set. By adopting this approach to the recommendation task, we hope to minimise the effects of inherent CF problems such as sparsity and latency [?].
Figure 1 below, outlines the architecture of this adaptive system. Data is sent to a classification module, which records the salient features of the dataset. Each of the recommendation techniques in the system then generates its own recommendations on the data. The recommendations from each technique are passed to the analysis module which performs accuracy testing on the recommendations, using train-test sets from the data. The performance of each algorithm, and the associated dataset classification is stored in an SQL database. A regression component then interpolates this data and outputs a regression function which can predict algorithm performance on a new dataset based only on its classification metrics.

1.1 Regression

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

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

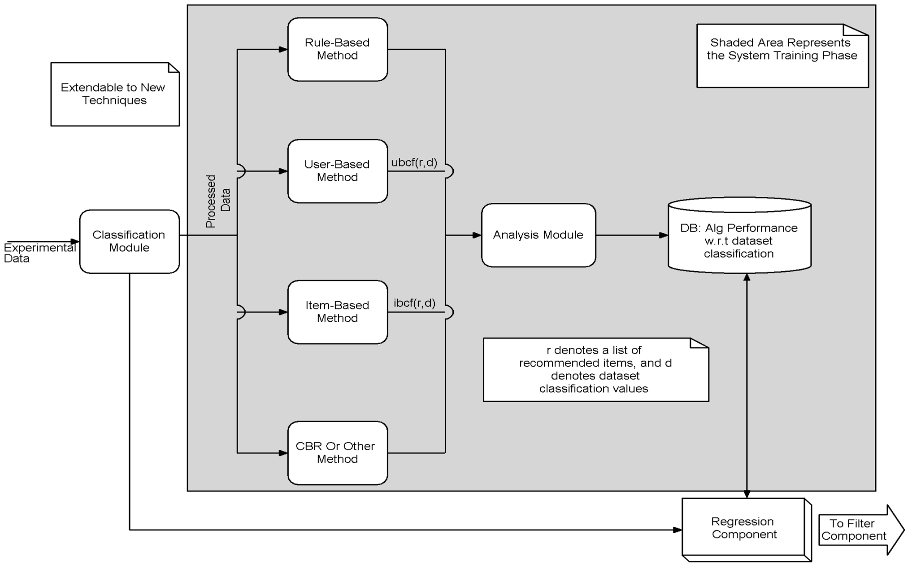

Fig. 1. AdRec System Architecture.

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.

2 Experimental Procedure

We classify our datasets based on a set of their salient features, including user-item ratio, rating distribution, data type, and sparsity of the data. We run performance tests on Item-Based Collaborative Filtering (IBCF) [**?**] [**?**], Pure Collaborative Filtering(UBCF) [**?**] [**?**], and an Association-Rule Based Filtering algorithm (RBCF) [**?**] [**?**] and a base-line classification algorithm: Zero-r, using four experimental datasets: PTV, Movielens, Jester and EachMovie.
Each of our recommendation algorithms differs in the manner in which it gathers similarity information about users. IBCF compiles a model of item similarity, based on the number of users who have co-rated each pair of items. UBCF is based on a model of user similarity, which is acquired by forming peergroups of users with high overlap of items in their profiles. The RBCF algorithm is similar to IBCF, except it employs data mining techniques (the Apriori association rule algorithm) to form associations between items, which are then used in an IBCF-like model of

item-item similarity. These rules take the form: $A \Longrightarrow B$, with associated support and confidence levels.
Our tests consider k-nearest neighbours, and density of the training set. For the former, we vary the value of k in intervals of 10 from 0 to 100, and note the predictive accuracy of our algorithm using standard leave-one-out analysis techniques. Our dataset density testing involves a similar predictive accuracy evaluation, this time varying the test-train ratio of the data from 10 to 90 percent, while maintaining k at its most optimal value of 30.

3 Results and Conclusion

Based on the results of our performance analysis, and the values each dataset produces for the classification metrics, we develop a multi-variable linear regression function for the purpose of predicting the suitability of a particular recommendation algorithm to a dataset previously unseen by the system, (The SmartRadio music ratings dataset). We plot a response-surface graph of the predictive accuracy of each recommendation algorithm with respect to the classification metrics for our different datasets.
Using this graph and the new datasets classification values, we aim to, and succeed in predicting the best performing algorithm for the new dataset. Our empirical results show that there is a significant difference in the relative performance of these algorithms over our four data platforms, and that the resulting regression function not only correctly predicted the best-performing algorithm for the new dataset, but also the second best performer. In this work we have shown how a meta-learning approach to designing a recommender system is beneficial to the end-user in that they receive the best available recommendations for their data, and to the system developer, since the decision as to what recommendation strategy to employ is completely automated, saving time and extensive research.