A

The Application of Data-Mining to Recommender Systems

J. Ben Schafer
University of Northern Iowa, USA

INTRODUCTION

In a world where the number of choices can be overwhelming, recommender systems help users find and evaluate items of interest. They connect users with items to "consume" (purchase, view, listen to, etc.) by associating the content of recommended items or the opinions of other individuals with the consuming user's actions or opinions. Such systems have become powerful tools in domains from electronic commerce to digital libraries and knowledge management. For example, a consumer of just about any major online retailer who expresses an interest in an item – either through viewing a product description or by placing the item in his "shopping cart" – will likely receive recommendations for additional products. These products can be recommended based on the top overall sellers on a site, on the demographics of the consumer, or on an analysis of the past buying behavior of the consumer as a prediction for future buying behavior. This paper will address the technology used to generate recommendations, focusing on the application of data mining techniques.

BACKGROUND

Many different algorithmic approaches have been applied to the basic problem of making accurate and efficient recommender systems. The earliest "recommender systems" were content filtering systems designed to fight information overload in textual domains. These were often based on traditional information filtering and information retrieval systems. Recommender systems that incorporate information retrieval methods are frequently used to satisfy ephemeral needs (short-lived, often one-time needs) from relatively static databases. For example, requesting a recommendation for a book preparing a sibling for a new child in the family. Conversely, recommender systems that incorporate information-filtering methods are frequently used to satisfy persistent information (long-lived, often frequent, and specific) needs from relatively stable databases in domains with a rapid turnover or frequent additions. For example, recommending AP stories to a user concerning the latest news regarding a senator's re-election campaign.

Without computers, a person often receives recommendations by listening to what people around him have to say. If many people in the office state that they enjoyed a particular movie, or if someone he tends to agree with suggests a given book, then he may treat these as recommendations. Collaborative filtering (CF) is an attempt to facilitate this process of "word of mouth." The simplest of CF systems provide generalized recommendations by aggregating the evaluations of the community at large. More personalized systems (Resnick & Varian, 1997) employ techniques such as user-to-user correlations or a nearest-neighbor algorithm.

The application of user-to-user correlations derives from statistics, where correlations between variables are used to measure the usefulness of a model. In recommender systems correlations are used to measure the extent of agreement between two users (Breese, Heckerman, & Kadie, 1998) and used to identify users whose ratings will contain high predictive value for a given user. Care must be taken, however, to identify correlations that are actually helpful. Users who have only one or two rated items in common should not be treated as strongly correlated. Herlocker et al. (1999) improved system accuracy by applying a significance weight to the correlation based on the number of co-rated items.

Nearest-neighbor algorithms compute the distance between users based on their preference history. Distances vary greatly based on domain, number of users, number of recommended items, and degree of co-rating between users. Predictions of how much a user will like an item are computed by taking the weighted average of the opinions of a set of neighbors for that item. As applied in recommender systems, neighbors are often generated online on a query-by-query basis rather than through the off-line construction of a more thorough model. As such, they have the advantage of being able to rapidly incorporate the most up-to-date information, but the search for neighbors is slow in large databases. Practical algorithms use heuristics to search for good neighbors and may use opportunistic sampling when faced with large populations.

Both nearest-neighbor and correlation-based recommenders provide a high level of personalization in their recommendations, and most early systems using these techniques showed promising accuracy rates. As such, CF-based systems have continued to be popular in recommender applications and have provided the benchmarks upon which more recent applications have been compared.

DATA MINING IN RECOMMENDER APPLICATIONS

The term data mining refers to a broad spectrum of mathematical modeling techniques and software tools that are used to find patterns in data and user these to build models. In this context of recommender applications, the term data mining is used to describe the collection of analysis techniques used to infer recommendation rules or build recommendation models from large data sets. Recommender systems that incorporate data mining techniques make their recommendations using knowledge learned from the actions and attributes of users. These systems are often based on the development of user profiles that can be persistent (based on demographic or item "consumption" history data), ephemeral (based on the actions during the current session), or both. These algorithms include clustering, classification techniques, the generation of association rules, and the production of similarity graphs through techniques such as Horting.

Clustering techniques work by identifying groups of consumers who appear to have similar preferences. Once the clusters are created, averaging the opinions of the other consumers in her cluster can be used to make predictions for an individual. Some clustering techniques represent each user with partial participation in several clusters. The prediction is then an average across the clusters, weighted by degree of participation. Clustering techniques usually produce less-personal recommendations than other methods, and in some cases, the clusters have worse accuracy than CF-based algorithms (Breese, Heckerman, & Kadie, 1998). Once the clustering is complete, however, performance can be very good, since the size of the group that must be analyzed is much smaller. Clustering techniques can also be applied as a "first step" for shrinking the candidate set in a CF-based algorithm or for distributing neighbor computations across several recommender engines. While dividing the population into clusters may hurt the accuracy of recommendations to users near the fringes of their assigned cluster, pre-clustering may be a worthwhile trade-off between accuracy and throughput.

Classifiers are general computational models for assigning a category to an input. The inputs may be vectors of features for the items being classified or data about relationships among the items. The category is a domain-specific classification such as malignant/benign for tumor classification, approve/reject for credit requests, or intruder/authorized for security checks. One way to build a recommender system using a classifier is to use information about a product and a customer as the input, and to have the output category represent how strongly to recommend the product to the customer. Classifiers may be implemented using many different machine-learning strategies including rule induction, neural networks, and Bayesian networks. In each case, the classifier is trained using a training set in which ground truth classifications are available. It can then be applied to classify new items for which the ground truths are not available. If subsequent ground truths become available, the classifier may be retrained over time.

For example, Bayesian networks create a model based on a training set with a decision tree at each node and edges representing user information. The model can be built off-line over a matter of hours or days. The resulting model is very small, very fast, and essentially as accurate as CF methods (Breese, Heckerman, & Kadie, 1998). Bayesian networks may prove practical for environments in which knowledge of consumer preferences changes slowly with respect to the time needed to build the model but are not suitable for environments in which consumer preference models must be updated rapidly or frequently.

Classifiers have been quite successful in a variety of domains ranging from the identification of fraud

Related Content