The Application of Data-Mining to Recommender Systems

J. Ben Schafer, Ph.D.
University of Northern Iowa

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 article 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 and 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 et al, 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 offline 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 et al., 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 et al., 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 and credit risks in financial transactions to medical diagnosis to intrusion detection. Good et al. (1999) implemented induction-learned feature-vector classification of movies and compared the classification with CF recommendations; this study found that the classifiers did not perform as well as CF, but that combining the two added value over CF alone.

One of the best-known examples of data mining in recommender systems is the discovery of association rules, or item-to-item correlations (Sarwar et. al, 2001). These techniques identify items frequently found in “association” with items in which a user has expressed interest. Association may be based on co-purchase data, preference by common users, or other measures. In its simplest implementation, item-to-item correlation can be used to identify "matching items" for a single item, such as other clothing items that are commonly purchased with a pair of pants. More powerful systems match an entire set of items, such as those in a customer's shopping cart, to identify appropriate items to recommend. These rules can also help a merchandiser arrange products so that, for example, a consumer purchasing a child's handheld video game sees batteries nearby. More sophisticated temporal data mining may suggest that a consumer who buys the video game today is likely to buy a pair of earplugs in the next month.

Item-to-item correlation recommender applications usually use current interest rather than long-term customer history, which makes them particularly well suited for ephemeral needs such as recommending gifts or locating documents on a topic of short lived interest. A user merely needs to identify one or more "starter" items to elicit recommendations tailored to the present rather than the past.

Association rules have been used for many years in merchandising, both to analyze patterns of preference across products, and to recommend products to consumers based on other products they have selected. An association rule expresses the relationship that one product is often purchased along with other products. The number of possible association rules grows exponentially with the number of products in a rule, but constraints on confidence and support, combined with algorithms that build association rules with itemsets of n items from rules with n-1 item itemsets, reduce the effective search space. Association rules can form a very compact representation of preference data that may improve efficiency of storage as well as performance. They are more commonly used for larger populations rather than for individual consumers, and they, like other learning methods that first build and then apply models, are less suitable for applications where knowledge of preferences changes rapidly. Association rules have been particularly successfully in broad applications such as shelf layout in retail stores. By contrast, recommender systems based on CF techniques are easier to implement for personal recommendation in a domain where consumer opinions are frequently added, such as on-line retail.

In addition to use in commerce, association rules have become powerful tools in recommendation applications in the domain of knowledge management. Such systems attempt to predict which web page or document can be most useful to a user. As Géry (2003) writes "The problem of finding Web pages visited together is similar to finding associations among itemsets in transaction databases. Once transactions have been identified, each of them could represent a basket, and each web resource an item." Systems built on this approach have been demonstrated to produce both high accuracy and precision in the coverage of documents recommended (Geyer-Schultz et. al, 2002).

Horting is a graph-based technique in which nodes are users, and edges between nodes indicate degree of similarity between two users (Wolf et al. 1999). Predictions are produced by walking the graph to nearby nodes and combining the opinions of the nearby users. Horting differs from collaborative filtering as the graph may be walked through other consumers who have not rated the product in question, thus exploring transitive relationships that traditional CF algorithms do not consider. In one study using synthetic data, Horting produced better predictions than a CF-based algorithm (Wolf et al. 1999).

FUTURE TRENDS

As data mining algorithms have been tested and validated in their application to recommender systems, a variety of promising applications have evolved. In this section we will consider three of these applications – meta-recommenders, social data mining systems, and temporal systems that recommend when rather than what.

Meta-recommenders are systems that allow users to personalize the merging of recommendations from a variety of recommendation sources employing any number of recommendation techniques. In doing so, these systems let users take advantage of the strengths of each different recommendation method. The SmartPad supermarket product recommender system (Lawrence et. al, 2001) suggests new or previously unpurchased products to shoppers creating shopping lists on a personal digital assistant (PDA). The SmartPad system considers a consumer's purchases across a store's product taxonomy. Recommendations of product subclasses are based upon a combination of class and subclass associations drawn from information filtering and co-purchase rules drawn from data mining. Product rankings within a product subclass are based upon the products' sales rankings within the user's consumer cluster, a less personalized variation of collaborative filtering. MetaLens (Schafer et. al, 2002) allows users to blend content requirements with personality profiles to allow users to determine which movie they should see. It does so by merging more persistent and personalized recommendations, with ephemeral content needs such as the lack of offensive content or the need to be home by a certain time. More importantly, it allows the user to customize the process by weighting the importance of each individual recommendation.

While a traditional CF-based recommender typically requires users to provide explicit feedback, a social data mining system attempts to mine the social activity records of a community of users to implicitly extract the importance of individuals and documents. Such activity may include Usenet messages, system usage history, citations, or hyperlinks. TopicShop (Amento et. al, 2003) is an information workspace which allows groups of common websites to be explored, organized into user defined collections, manipulated to extract and order common features, and annotated by one or more users. These actions on their own may not be of large interest, but the collection of these actions can be mined by TopicShop and redistributed to other users to suggest sites of general and personal interest. Agrawal et. al (2003) explored the threads of newsgroups to identify the relationships between community members. Interestingly, they concluded that due to the nature of newsgroup postings – users are more likely to respond to those with whom they disagree – "links" between users are more likely to suggest that users should be placed in differing partitions rather than the same partition. Although this technique has not been directly applied to the construction of recommendations, such an application seems a logical field of future study.

Although traditional recommenders suggest what item a user should consume they have tended to ignore changes over time. Temporal recommenders apply data mining techniques to suggest when a recommendation should be made or when a user should consume an item. Adomavicius et. al (2001) suggest the construction of a recommendation warehouse which stores ratings in a hypercube. This multidimensional structure can store data on not only the traditional user and item axes, but also for additional profile dimensions such as time. Through this approach, queries can be expanded from the traditional "what items should we suggest to user X" to "at what times would user X be most receptive to recommendations for product Y." Hamlet (Etzioni et. al, 2003) is designed to minimize the purchase price of airplane tickets. Hamlet combines the results from time series analysis, Q-learning, and the Ripper algorithm to create a multi-strategy data-mining algorithm. By watching for trends in airline pricing and suggesting when a ticket should be purchased, Hamlet was able to save the average user 23.8% when savings was possible.

CONCLUSION

Recommender Systems have emerged as powerful tools for helping users find and evaluate items of interest. These systems use a variety of techniques to help users identify the items that best fit their tastes or needs. While popular CF-based algorithms continue to produce meaningful, personalized results in a variety of domains, data mining techniques are increasingly being used in both hybrid systems, to improve recommendations in previously successful applications, and in stand-alone recommenders, to produce accurate recommendations in previously challenging domains. The use of data mining algorithms has also changed the types of recommendations as applications move from recommending what to consume to also recommending when to consume. While recommender systems may have started as largely a passing novelty, they clearly appear to have moved into a real and powerful tool in a variety of applications, and that data mining algorithms can be and will continue to be an important part of the recommendation process.

REFERENCES

Adomavicius, G., & Tushilin, A. (2001). Extending Recommender Systems: A Multidimensional Approach. IJCAI-01 Workshop on Intelligent Techniques for Web Personalization (ITWP'2001).

Agrawal R., Rajagopalan, S., Srikant, R., & Xu, Y. (2003). Mining newsgroups using networks arising from social behavior. Proceedings of the Twelfth World Wide Web Conference. pp. 529-535

Amento, B., Terveen, L., Hill, W., Hix, D., & Schulman, R (2003). Experiments in Social Data Mining: The TopicShop System, in ACM Transactions on Computer-Human Interaction, 10, 1, pp 54-85.

Breese, J., Heckerman, D., & Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98), pp 43-52.

Etzioni, O., Knoblock, C.A., Tuchinda, R., & Yates, A. (2003). To Buy or Not to Buy: Mining Airfare Data to Minimize Ticket Purchase Price, Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 119-128.

Géry, M., & Haddad, H. (2003). Evaluation of Web Usage Mining approaches for user's next request prediction. Fifth International Workshop on Web Information and Data Management. pp. 74-81.

Geyer-Schulz, A., & Hahsler, M. (2002). Evaluation of Recommender Algorithms for an Internet Information Broker based on Simple Association Rules and on the Repeat-Buying Theory. *Fourth WEBKDD Workshop: Web Mining for Usage Patterns & User Profiles*, pp. 100-114.

Good, N., Schafer, J.B., Konstan, J.A., Borchers, A., Sarwar, B., Herlocker, J., & Riedl, J. (1999). Combining Collaborative Filtering with Personal Agents for Better Recommendations, Proceedings of Sixteenth National Conference on Artificial Intelligence (AAAI-99), AAAI Press. pp 439-446.

Herlocker, J., Konstan, J.A., Borchers, A., and Riedl, J. (1999). An Algorithmic Framework for Performing Collaborative Filtering. Proceedings of the 1999 Conference on Research and Development in Information Retrieval.

Lawrence, R.D., Almasi, G.S., Kotlyar, V., Viveros, M.S., & Duri, S.S. (2001). Personalization of Supermarket Product Recommendations. Data Mining and Knowledge Discovery 5(1/2) pp.11-32.

Lin, W., Alvarez, S.A., & Ruiz, C (2002). Efficient adaptive-support association rule mining for recommender systems. Data Mining and Knowledge Discovery, 6(1) pp. 83-105.

Resnick, P. & Varian, H.R. (1997) Communications of the Association of Computing Machinery Special issue on Recommender Systems, 40(3):56–89.

Sarwar, B., Karypis, G., Konstan, J.A., & Reidl, J. (2001). Item-based Collaborative Filtering Recommendation Algorithms. Proceedings of the Tenth International Conference on World Wide Web, pp. 285 - 295

Schafer, J.B., Konstan, J.A., & Riedl, J. (2001). E-Commerce Recommendation Applications. Data Mining and Knowledge Discovery 5(1/2) pp.115-153.

Schafer, J.B., Konstan, J.A., & Riedl, J. (2002). Meta-recommendation Systems: User-controlled Integration of Diverse Recommendations. Proceedings of the Eleventh Conference on Information and Knowledge (CIKM-02). pp. 196-203.

Shoemaker, C., & Ruiz, C. (2003) Association Rule Mining Algorithms for Set-Valued Data. Lecture Notes in Computer Science Vol. 2690. pp. 669-676.

Wolf, J., Aggarwal, C., Wu, K-L., & Yu, P. (1999). Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering. In Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, San Diego, CA.

Terms and Definitions

Association Rules: Used to associate items in a database sharing some relationship (e.g. co-purchase information). Often takes the for "if this, then that" such as "If the customer buys a handheld videogame then the customer is likely to purchase batteries."

Collaborative Filtering: Selecting content based on the preferences of people with similar interests.

Meta-recommenders: Provide users with personalized control over the generation of a single recommendation list formed from the combination of rich recommendation data from multiple information sources and recommendation techniques.

Nearest-Neighbor Algorithm: A recommendation algorithm that calculates the distance between users based on the degree of correlations between scores in the users' preference histories. Predictions of how much a user will like an item are computed by taking the weighted average of the opinions of a set of nearest neighbors for that item.

Recommender Systems: Any system that provides a recommendation, prediction, opinion, or user-configured list of items that assists the user in evaluating items.

Social Data-Mining: Analysis and redistribution of information from records of social activity such as newsgroup postings, hyperlinks, or system usage history.

Temporal Recommenders: Recommenders that incorporate time into the recommendation process. Time can be either an input to the recommendation function, or the output of the function.