Integrating Knowledge-based and Collaborative-filtering Recommender Systems

Robin Burke

Recommender.com, Inc.
and
Information and Computer Science
University of California, Irvine
Irvine, CA 92697
burke@ics.uci.edu

Abstract

Knowledge-based and collaborative-filtering recommender systems facilitate electronic commerce by helping users find appropriate products from large catalogs. This paper discusses the strengths and weaknesses of both techniques and introduces the possibility of a hybrid recommender system that combines the two approaches. An approach is suggested in which knowledge-based techniques are used to bootstrap the collaborative filtering engine while its data pool is small, and the collaborative filter is used as a post-filter for the knowledge-based recommender.

Collaborative Filtering

Collaborative filtering recommender system are a widely-accepted technique in electronic commerce. (See [Resnick & Varian, 1997] and other articles in that special issue. A recent survey is found in [Maes, Guttman & Moukas, 1999]. See also [Goldberg et al. 1992] and [Resnick, et al. 1994].) These systems aggregate data about customers' purchasing habits or preferences and make recommendations to other users based on similarity in overall patterns. For example, in the Ringo music recommender system (Shardanand & Maes, 1995), users who had expressed their musical preferences by rating various artists and albums could get suggestions of other groups and recordings that others with similar preferences also liked.

As a collaborative filtering system collects more ratings from more users, the probability increases that someone in the system will be a good match for any given new user. This beneficial property also has its downside, however. A collaborative filtering system must be initialized with a large amount of data, because a system with a small base of ratings is unlikely to be very useful. Further, the accuracy of the system is very sensitive to the number of rated items that can be associated with a given user (Shardanand & Maes, 1995). These factors contribute to a "ramp-up" problem: until there is a large number of users whose habits are known, the system cannot be useful for most users, and until a sufficient number of rated items has been collected, the system cannot be useful for a particular user.[1]

Another problem with collaborative filtering systems might be called the "banana" problem. Bananas are a frequently-purchased item in most American grocery stores, and the odds are high that any given market basket will contain bananas. A naive recommender system working from market basket data will always recommend bananas, simply because they are highly correlated with everything. Because the system has no notion of what foods ought to go together, it cannot screen out such suggestions.

These drawbacks are not significant for some large e-commerce sites, such as Amazon.com, with a very large customer base, and a large and diverse product line that lends itself to multi-item purchases. A more difficult challenge is presented for a product such as an automobile that is bought much less frequently and one at a time. For an automobile, a home loan or any other infrequently-purchased item, the system will not be able to use market-basket or purchase history to make recommendations. A recommender system would never be able to say "people who bought a Geo Metro also bought a Ford Escort," because that is not how people buy cars.

Knowledge-based recommender systems

What a recommender system for a car or other similar product must do is get information about users' preferences: Why are they buying a car? Is comfort or fuel economy more important? Based on such information, the system can pursue a knowledge-based approach to generating a recommendation, by reasoning about what products meet the user's requirements. The PersonalLogic recommender system offers a dialog that effectively walks the user down a discrimination tree of product features.[2] Other systems have adapted quantitative decision support tools for this task (Bhargava, Sridhar & Herrick, 1999). Another

[1] The need to maintain user-identified logs of preferences and purchases also raises privacy concerns for collaborative filtering systems.

[2] <URL: http://www.personallogic.com/>

Technique	Pluses	Minuses
Knowledge-based	A. No ramp-up required B. Detailed qualitative preference feedback (in FindMe systems) C. Sensitive to short-term variance (drift)	H. Knowledge engineering. I. Suggestion ability is static.
Collaborative filtering	D. Can identify niches precisely. E. Domain knowledge not needed. F. Quality improves over time. G. Personalized recommendations.	J. Quality dependent on large historical data set. K. Subject to statistical anomalies in data. L. Reacts slowly to drift
Ideal Hybrid	A, B, C, D, F, G	H

Table 1: Tradeoffs between knowledge-based and collaborative-filtering recommender systems.

class of systems draws from research in case-based reasoning. The restaurant recommender Entree[1] (Burke, Hammond & Cooper, 1996) makes its recommendations by finding restaurants in a new city similar to restaurants the user knows and likes. The system allows users to navigate by stating their preferences with respect to a given restaurant, refining their search criteria.

A knowledge-based recommender system avoids some of the drawbacks mentioned above: it does not have a ramp-up problem since its recommendations do not depend on a base of user ratings. It does not have to gather information about a particular user because its similarity judgements are independent of individual tastes. Because its recommendations are based on knowledge of the product domain, it is immune to statistical anomalies in market baskets.

FindMe systems

Entree is an example of what is known as a FindMe system (Burke, Hammond & Young, 1997). FindMe systems have the following distinguishing characteristics as recommender systems.

- They are primarily example-based. Users can easily find new products similar to ones with which they are already familiar.
- They allow the user to critique a suggestion, and try to find similar products that satisfy the critique.
- They rank products based on the expected goals of users, such as cost-effectiveness, and return a small number of highly-rated products.

FindMe systems effectively offer a web of products, in which the links are product-dependent critiques, such as "Less $$" or "More traditional" cuisine, in the case of restaurants. For example, if the user finds a restaurant that looks good but finds it is too expensive, he can select the "Less $$" button and be directed to other restaurants that are similar, but lower cost.

A particular benefit of FindMe systems is that they enable relatively painless gathering of preference information without requiring that users make all of their criteria explicit. Rather than requiring the user to input his or her preferences as a starting point, FindMe systems let the user browse through a catalog using qualitative ratings as navigation aids. Each navigation step informs the system about the user's preferences at a finer grain of detail than a binary "buy" decision can, and a user is likely to make several (typically 3 in Entree) such navigation steps while using the system, increasing the amount of information that can be gathered.

Another benefit of FindMe systems is that performance does not suffer if the user's interests drift. A movie buff who usually likes classic film noir will find a well-tuned collaborative filtering system less useful when he seeks out movies for his children. A FindMe system works from whatever starting point it is given and reacts to the user's preferences at that time.

Knowledge Engineering

Knowledge-based recommender systems do require knowledge engineering with all of its attendant difficulties. For a system to make good recommendations, it must understand what features of products matter. It must have access to a product database in which those features are readily discernable or at least inferable. In FindMe systems, we have found this necessity substantial but not onerous. For example, the VintageExchange FindMe recommender system for wines (Burke, 1999) required approximately one person-month of knowledge engineering effort.

Both Entree and VintageExchange also required significant data cleaning and natural language processing to render database entries useful. In Entree, cross-reference data (such lists of all restaurants offering Sunday brunch) were inverted to create per-restaurant feature sets. Restaurant reviews were also processed for key words and phrases.

Combining recommendation techniques

Table 1 contrasts the collaborative filtering and knowledge-based approaches, identifying the positive and negative aspects of each. The third row suggests what might be

[1] <URL: http://infolab.ils.nwu.edu/entree/>

achieved in an ideal hybrid that combines the techniques. Despite the necessary investment in knowledge engineering, such a hybrid offers good performance even with little or no user data, and the benefits of collaborative filtering as data is collected. The possible synergy with FindMe systems appears particularly promising, since these systems, through preference-based browsing, permit the collection of detailed user ratings even for rarely-purchased items like automobiles.

Achieving integration of these techniques is an interesting challenge problem for the field of recommender systems. There are numerous possibilities to be explored. At Recommender.com, we are currently developing FindMe technology for commercial applications and are exploring one such hybrid approach, as outlined in the rest of this paper.

Cascaded recommendations

A FindMe system essentially performs an alphabetic sort over the space of products, ranking them according to a set of prioritized similarity metrics M_0-M_n. (For a full description of the FindMe algorithm, see [Burke, Hammond & Young, 1997].) The result is an ordered set of buckets B_0-B_m, equivalence classes of products that are considered equally good suggestions based on the user's input. A fixed number of the topmost items are returned to the user. Often, similarity measures fail to discriminate the returned items completely. The system must arbitrarily return 10 items, for example, out of a bucket B_0 of size 15 or 20.

In such a case, we consider the result "under-discriminated." The complete elimination of under-discriminated results in FindMe systems is difficult because it requires the addition of more similarity metrics, with attendant knowledge-engineering tasks. Collaborative filtering, however, can add additional discrimination without requiring knowledge engineering.

At an abstract level, there are four different kinds of preference information that a FindMe system collects from users:

Entry point: The item that the user chooses as a starting point can be considered a strongly positive preference, since the user is looking for something similar to it.

Ending point: The final selection or buying decision can also be considered a positive rating.

Tweaking: When a user critiques a returned item and moves on to something different, we can consider this a negative rating.

Browsing: If the user navigates to other items in the returned set, we can consider this a weak negative rating: if the user truly liked the item he or she would probably not browse further.

These heuristics are somewhat weak, since we sometimes find users who are exploring the system to see what it can do, applying tweaks not to get a specific recommendation, but to see what will come back. However, we believe they form a reasonable first approximation of the notion of a "rating" within the FindMe context.

Let us assume that, in addition to the FindMe recommender system, there is also a collaborative filtering engine available, where ratings are obtained for each item that the user encounters in a FindMe session using the simplified rating scheme given above, and applying standard correlation techniques.

We refer to this as a single-scale filter or SS filter. Using it, we can derive similar ratings from all of the user's action, look for similarities across users, and return previously-unseen items in a standard collaborative filtering manner.

The operation of the SS filter is likely to be weak if it starts with a small amount of data, so we would not want to present its suggestions directly to users. However, there is little risk of making a bad suggestion if we use only those ratings of the items in the topmost under-discriminated bucket B_0. We can go though each item b_j in such a bucket and attempt to classify it into high or low preference category based on the user's interaction so far. These categories can then to be used to discriminate the contents of B_0. Intuitively, all other things being equal, we will prefer to recommend b_j, if the acceptance of that recommendation would make the user similar to some set of other users, and prefer not to recommend an item if similar users navigated away from it in the past. In the worst case, if the ratings from the SS filter are uninformative, we will still be selecting items that are equally similar as far as our knowledge-based system is concerned, so the technique can do no harm.

This would be a cascade from FindMe to the SS filter, where the collaborative filter is only used after the knowledge-based system has done its work. This design would potentially get us part of the way toward the ideal hybrid discussed above. A system designed in this way would have all of the benefits of the FindMe system, but its quality would improve over time in a personalized way.

Consider the following example: Alice connects to a version of Entree that includes the collaborative filtering component. She registers as a new user, and starts browsing for Chicago restaurants by entering the name of her favorite restaurant at home, *Greens Restaurant* in San Francisco. Greens is characterized as serving "Californian Vegetarian" cuisine. The top recommendation is *302 West*, which serves "Californian Seafood." It turns out that Alice is, in fact, a vegetarian, so she critiques the system's cuisine choice and moves back towards vegetarian recommendations.

After the system has built up a bigger user base, another new user Ben approaches the system with the same starting point: *Greens*. Since the recommendation given to Alice was under-discriminated, and her feedback and that of other users allows the system to more fully discriminate Ben's recommendation, and return *Jane's*, a vegetarian restaurant, preferring it over *302 West*.

This thought experiment suggests that a cascade using both knowledge-based and collaborative-filtering techniques may produce a recommender system with many of the characteristics of an ideal hybrid. Initial suggestions are good, since there is a knowledge-base to rely on. As the system's database of ratings increases, it can move beyond the knowledge base to characterize users more precisely. Because the knowledge base is always present, users are not trapped by their past behavior. If Alice decides to stop being a vegetarian, she will be able to get recommendations for steakhouses.

Collaborative filtering of critiques

The technique discussed above takes into account only a coarse representation of a user's search through the FindMe system and does not take into account the specific nature of the critiques that the user makes of the system's suggestions. We can use these critiques by thinking of them as multi-dimensional ratings on different scales. We split up a user's ratings into segments: those restaurants that the user thought were too expensive, too quiet, too conservative, etc.

Our comparison of users can thus become more global – identifying others who not only liked the same things, but also disliked the same things for the same reasons. We anticipate improved recommendations against this data, discovering for example that user who thought *Yoshi's Cafe* was too expensive often liked *Lulu's*. We anticipate strong clusters in the data: for example, "cheapskates" and "epicures" who would regularly apply "cheaper" and "better" tweaks, respectively. However, this technique increases the sparseness of the rating data, so only empirical evaluation will determine if it will provide an improvement over the rougher single-scale approach.

The Entree restaurant guide has been under continuous public operation since August 1996. The HTTP logs of its operation are forming the basis for our first experiments with these ideas. The logs record interactions with over 20,000 unique users (as identified by IP address) rating about 1,000 restaurants.

Conclusion

Knowledge-based and collaborative-filtering recommender systems each have distinctive properties that lend themselves to electronic commerce. We have suggested that the integration of these techniques is an important unsolved problem in the area of recommender systems. We have shown one approach to the construction of a hybrid system, within the context of FindMe architecture, and shown how synergy between the techniques might be achieved.

The hybrid approaches discussed here do not capitalize on the full power of collaborative filtering, which, in its pure form, permits the discovery of niche groups of consumers who share tastes. Our approach to filtering would only make itself felt in cases where the existing knowledge-based system was discriminating poorly. Further research will be needed to explore other hybrid recommendation schemes.

References

Bhargava, H. K., Sridhar, S. and Herrick, C. 1999. Beyond Spreadsheets: Tools for Building Decision Support Systems. *IEEE Computer*, 32(3) 31-39.

Burke, R, 1999. The Wasabi Personal Shopper: A Case-Based Recommender System. Submitted to the 11th Annual Conference on Innovative Applications of Artificial Intelligence.

Burke, R., Hammond, K. & Cooper, E. 1996. Knowledge-based navigation of complex information spaces. In *Proceedings of the 13th National Conference on Artificial Intelligence,* 462-468, Menlo Park, CA: AAAI Press.

Burke, R., Hammond, K., and Young, B. The FindMe Approach to Assisted Browsing. *IEEE Expert*, 12(4), pages 32-40, 1997.

Goldberg, D., Nichols, D., Oki, B. M., Terry, D. 1992. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12) 61-70.

Maes, P., Guttman, R. H., and Moukas, A. G. 1999. Agents that buy and sell. Communications of the ACM, 42(3), 81-91.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. GroupLens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the conference on Computer supported cooperative work,* 175-186. New York: ACM Press.

Resnick, P. and Varian, H. R. 1997. Recommender systems. *Communications of the ACM*, 40(3) 56-58.

Shardanand, U. and Maes, P. 1995. Social information filtering algorithms for automating "word of mouth" In *CHI-95: Conference proceedings on Human factors in computing systems* , 210-217. New York: ACM Press.