Talk to my Agents: Research Issues in Combining Information Filtering Agents with Collaborative Filtering

J. Ben Schafer, Nathaniel Good
GroupLens Research Project
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
schafer@cs.umn.edu, good@cs.umn.edu
http://www.grouplens.org

ABSTRACT

As the Internet increases our ability to access information, it also increases the severity of information overload – the inability to find meaningful information among the slew of non-relevant information. Researchers have explored several methods for reducing information overload – each with their strengths and their weaknesses. Information Filtering (IF) extracts item content and makes recommendations based on matches with a user interest profile. Collaborative Filtering (CF) matches users with other users with similar tastes to theirs, and makes recommendations based on the opinions of others in these "neighborhoods."

This paper describes a series of experiments conducted by members of the GroupLens Research Project which show that a CF framework can be used to combine either generic filtering agents or personalized IF agents with the opinions of a community of users to produce better recommendations than either agents or users can produce alone. We have also shown that using CF to create a personal combination of a set of agents produces better results than either individual agents or other combination mechanisms. One key implication of these results is that users can avoid having to select among agents; they can use them all and let the CF framework select the best ones for them.

INTRODUCTION

As computer technology has advanced, the amount of information that systems can store has grown at an alarming rate. As the amount of information available grows, so does the difficulty for any one user to locate useful information, let alone comprehend it all. To assist in the challenges of information overload, recommender systems have been developed that attempt to link users and communities with the items/data that they will find useful. Collaborative filtering (CF) based recommender systems attempt to match a given end user with a neighborhood of "like-minded" users whose opinion of "undigested" items can be utilized to make predictions. For example, if User A has not seen "Star Wars" yet, the opinions of the members of his neighborhood can be used to make a prediction as to whether or not he should rent it. Information filtering (IF) based recommender systems on the other hand, do not know about the opinions of other users. Instead, they examine syntactic and semantic content of items to make predictions for a given end user. For example, an IF system might recommend "Star Wars" to User B because she has shown a tendency to like science fiction movies, movies directed by George Lucas and movies starring Harrison Ford.

In our recent work we have examined simple rule based filtering agents, personal information filtering agents, collaborative filtering, and mechanisms for combining them to produce better recommender systems. The following sections discuss three of these projects.

Using Filterbots to Improve Predictions

While collaborative filtering has been considered a success, two key problems have been identified in research and through commercial products. The **early-rater problem** points out that a CF system provides little to no value to the first user in his neighborhood to enter a rating for an item. CF systems depend on users who are willing to under take the "cost" of providing that initial rating without the "reward" of receiving many recommendations. The **sparsity problem** points out that for many environments users may only cover a small portion of items available. Usenet studies have shown a rating rate of about 1% in some areas; we can estimate that few people will have read and formed an opinion on even 1/10 of 1% of the over two million books available through the largest bookstores. In fact, this scarcity is a key motivator behind recommender systems – people do not want to have to dig through large numbers of items to find the few that are relevant to them.

Initial work by our group, (Sarwar, et. al. 1998) proposed reducing these problems through the use of *filterbots* – automated rating robots that evaluate items as they are added to the system. Three simple bots were created:

- SpellCheckerBot – rates articles based on the proportion of spelling errors in the article text. The higher the proportion the lower the rating.
- IncludedMsgBot – rates articles based on the percentage of text quoted from other articles. The higher the percentage the lower the rating.
- LengthBot – rates articles based on the length of the actual message (minus headers, signatures, and included text). The longer the length the lower the rating.

This work showed that each filterbot could participate as a member of a collaborative filtering system – prolific raters who never ask for recommendations. In doing so, the bot not only reduced the severity of the aforementioned problems – a filterbot "user" rates *every* article before a single human user gets involved – but helped users who agree with it by providing more ratings upon which recommendations could be made. The key advantages of this type of implementation is that no "broad based" decisions are made which help some users, but hinder others. For users who did not agree with the filterbot, the CF framework would notice a low preference correlation and not make use of their ratings.

Generating Personalized Recommendation Agents

Follow up work by our group, (Good, et. al. 1999) extended the filterbot concept in three key ways. First, we used a more intelligent set of filterbots, including learning agents that are personalized to an individual user. Second, we applied this work to small communities, including using CF to serve a single human user. Third, we evaluated the simultaneous use of multiple filterbots. In addition, we explored other combination mechanisms as alternatives to CF. Specifically, we looked at four key models:

- Pure collaborative filtering using the opinions of other community members
- A single personalized "agent" – a machine learning or syntactic filter
- A combination of many "agents"
- A combination of multiple agents and community member opinions

The project utilized three classes of filterbots. The first, GenreBots, consisted of 19 simple bots corresponding to the 19 genres provided at the Internet Movie Database (www.imdb.com). Each rated a movie a 5 if the movie matched the given bot's genre and a 3 otherwise. Furthermore, a personalized *Mega-GenreBot* was created for each user. using linear regression. The second class, Doppelganger Bots (DGBots) are personalized bots that create profiles of user preferences and generate predictions using IR/IF techniques; specifically, a modified TFIDF, based upon the content features of each movie. Three DGBots were created utilizing keyword information, cast information, and a combination of the two. Finally, RipperBot was created using Ripper, an inductive logic program created by William Cohen.

We identified five different strategies for combining agents: selecting one agent for each person, averaging the agents together, using regression to create a personal combination, generating a classification and regression tree (CART) create a personal combination, and using CF to create a personal combination (a single user and his agents only). For all but the first of these, we found it valuable to create two combinations: one that used all 19 GenreBots and one that used the Mega-GenreBot. Adding the 3 DGBots and RipperBot, we refer to these as 23-agent and 5-agent versions, respectively.

In the final model, Combination of Users and IF Agents, we used CF to combine the 23 agents and all 50 users. The method is identical to the CF combination of agents except that we also loaded the ratings for the other 49 users.

The most important results we found true were that a personalized combination of several agents provides better recommendations than a single agent, and that a personalized combination of several agents and community opinions provides better recommendations than either agents or user opinions alone. This was particularly true using our CF engine. In essence, these results suggest that an effective mechanism for producing high-quality recommendations is to throw in any available data and allow the CF engine to sort out which information is useful to each user. In effect, it becomes less important to invent a brilliant agent, instead we can simply invent a collection of useful ones. We should point out that these experiments tested the quality of the resulting recommender system, not the performance or economics of such a system. Current CF recommendation engines cannot efficiently handle "users" who rate all items and re-rate them frequently as they "learn." To take advantage of learning agents, these engines must be redesigned to accommodate "users" with dynamic rating habits. We are examining several different CF engine designs that could efficiently use filterbots.

We were also pleased, though somewhat surprised, to find that CF outperformed linear regression as a combining mechanism for agents. While linear regression should provide an optimal linear fit, it appears that CF's non-optimal mechanism actually does a better job avoiding overfitting the data when the number of columns approaches the number of rows. CF also has the advantage of functioning on incomplete (and indeed very sparse) data sets, suggesting that it retains its value as a useful combination tool whenever human or agents are unlikely to rate each item.

We were surprised by several of the results that we found. In particular, we discovered that either a single personalized agent or combination of agents could provide better recommendations than the opinions of a community of users. We clearly overestimated the value of collaborative filtering for a small community of 50 users. In retrospect, our expectations may have been built from our own positive experiences when starting CF systems with a small group of researchers and friends. Those successes may have been due in part to close ties among the users; we often had seen the same movies and many had similar tastes. Using real users resulted in real diversity which may explain the lower, and more realistic, value. Future work should both incorporate larger user sets and look explicitly at closer-knit communities to see whether a smaller but more homogeneous community would have greater benefits from collaborative filtering. We also were surprised by the results we achieved using Ripper. We were impressed by its accuracy, after extensive tuning, but dismayed by how close to random it was in distinguishing good from bad movies. We are still uncertain as to why RipperBot performs as it does, and believe further work is needed to understand why it behaves as it does and whether it would be possible to train it to perform differently.

Analyzing the Importance of Individual Agents in Combinations of Agents

For the most recent portion of this project, we wanted to explore the importance or "weights" assigned to each of the individual agents for a given user and combination methods. The most important result we found true is that the importance of a single agent within a personalized combination of agents differs among users. In other words, there is no single agent which is consistently ranked high among all users. While this is difficult to "prove statistically" analysis of several sets of figures confirm this observation.

At first observation we were surprised to notice that for a given user, the importance of a single agent varied between combination methods. It was our initial belief that, for example, if User C's CART makes its initial decision based on the value of the MegaGenreBot than in linear regression, the Beta with the largest absolute value[1] would be that which corresponds to the variable represented by the MegaGenreBot and in collaborative filtering, the highest correlation would exist between the user and the "user" represented by the MegaGenreBot.

Upon examination we discovered this was not the case. However, upon further study this result isn't necessarily surprising. While each of the methods attempts to find an optimal solution, the methods behind the generation of such solutions, and what the methods define as "optimal," changes radically from method to method. For example, CART considers "optimal" to be the tree such that the percentage of items classified correctly is as high as possible. The penalty for being "wrong" is the same regardless of "how wrong. Thus, an item whose correct classification is a 2-rating, is penalized the same whether it incorrectly classified as a "1" or a "5." Linear regression on the other hand is not as concerned about the frequency of "wrong" answers, as it is in reducing the distance between the wrong answer and the "correct" value. Thus, it is not surprising that the methods apply different weights to different features in their different efforts to define optimal.

An interesting pattern emerges when we examine the average "distance" between each of the pairs of methods over all users. The encouraging news is that collaborative filtering produces a model which is roughly equidistant between the other models. Thus, we can think of it as a happy medium. It escapes the rigidity of either of the other models. It is neither as "demanding" as linear regression (it does not "require" that all agents be utilized) nor as "economical" as CART (which tries to make decisions on as few pieces of information as possible to avoid overfitting).

The implications of these final observations are quite exciting. The fact that different combination methods weight agents differently suggests that the determination of which agents are "important" should not be taken lightly. The determination of the "single best" agent for a user or a group of users, is not a simple matter. The effectiveness of an agent is highly dependent on the data at hand. This suggests that several complementary methods which weight different agents differently will more likely to be able to compensate for variances in the data. If we simply depend on one agent, or even combination of agents, errors in the agents and/or the data supporting them will be more likely to increase the error of the system. However, it is interesting to note that if a single combination method were all that was utilized, that not only did CF produce the best results, but as a single

[1] The *absolute values* of the ßs need to be taken since a strongly negative beta does not indicate a non-important variable. On the contrary, it indicates a variable which strongly effects the overall prediction but in the negative direction (e.g. an item which the user strongly dislikes).

method it was the most "central." This may suggest that it would be less effected by variances in the data since it does not reach too far to one side.

Future Work

While this work answers a number of questions about the feasibility of combining information filtering with collaborative filtering, it raises as many questions as it answers.

First of all, our prior work was based on both agents and combinations which were relatively simple. It is worth pursuing further combinations of users and perhaps more advanced agents in recommender systems. For example, we make the assumption that the content in this domain can be best fit with *linear* regression. Linear regression assumes that there is little interaction between the agents in developing the overall prediction and although not equally weighted, a linear rise in the agents should produce a linear rise in the prediction. Although we have no immediate reason to doubt these assumptions, there is also no reason why we should assume it to be the case. Perhaps one of the non-linear regression methods would be more accurate.

Similarly, work needs to be done in the examination of the error associated with each agent within each regression. At this point in time we are assuming that each agent IS important in the overall model. A preliminary exploration suggests "retraining" regressions minus low ß variables will reduce model error without increasing predictive error.

Additionally, this work was limited to predictions generated from a small set of users and a single users agents. What are the implications of introducing even more users into the system? Will users who agree with each other also benefit from the opinions of each other's agents?

Finally, assuming that this "MegaCF" system is feasible and beneficial, what is the overall benefit of a given agent in a collaborative filtering community? If for every new user added to the CF system it is also necessary to add 5-24 *additional* "users" (the new user and her agents) then the overhead of maintaining a CF site rises at a much higher rate. It becomes important to identify which non-human users (agents) are beneficial to the community as a whole, and which agents do not contribute enough benefit to warrant the cost of their inclusion. Will it become necessary for designers of recommendation systems to consider the implementation of a "Darwinistic community" where an agent is removed from the system because its "cost" to the "society" exceeds its benefit? Anecdotally, it would be interesting to calculate for how many people someone else's agent beats out their own, personalized agent – i.e. for how many users is their correlation with their own RipperBot lower than their correlation with someone else's RipperBot?

REFERENCES

1. Basu C., Hirsh H., and Cohen, W.W. 1998. Using Social and Content-Based Information in Recommendation. In *Proceedings of the AAAI-98,*:AAAI Press.
2. Belkin, N., and Croft, B.W. 1992. Information Filtering and Information Retrieval: Two Sides of the Same Coin?. *Communications of the ACM* 35(12):29-38.
3. Breese, J. Heckerman, D., and Kadie, C. 1998. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, Madison, WI.
4. Cohen, W.W. 1995. Fast Effective Rule Induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, Lake Tahoe, CA.:AAAI Press.
5. Good, N., Schafer, J.B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., and Riedl, J., Combining Collaborative Filtering with Personal Agents for Better Recommendations. To be presented at AAAI-99.
6. Hill, W., Stead, L., Rosenstein, M., and Furnas, G., 1995. Recommending and Evaluating Choices in a Virtual Community of Use. In *Proceedings of ACM CHI'95*, 194-201. Denver, CO.: ACM.
7. Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., and Riedl, J. 1997 Applying Collaborative Filtering to Usenet News. *Communications of the ACM* 40(3):77-87.
8. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews.. In Proceedings of 1994 Conference on Computer Supported Collaborative Work, 175-186.: ACM.
9. Saltan, G., Buckley, C. 1987. Term Weighting Approaches in Automatic Text Retrieval., Technical Report, Dept. of Computer Science, Cornell Univ.
10. Sarwar, B.M., Konstan, J.A., Borchers, A., Herlocker, J.L., Miller, B.N., and Riedl, J. 1998. Using Filtering Agents to Improve Prediction Quality in the Grouplens Research Collaborative Filtering System. In *Proceedings of CSCW '98*, Seattle, WA.: ACM.
11. Terveen, L., Hill, W., Amento, B., McDonald, D., Creter J. 1997. PHOAKS: A System for Sharing Recommendations. Communications of the ACM 40(3):59-62.