DynamicLens: A Dynamic User-Interface for a Meta-Recommendation System

J. Ben Schafer
Department of Computer Science
University of Northern Iowa
Cedar Falls, IA 50614-0507 USA
+1 319 273 2187
schafer@cs.uni.edu

ABSTRACT
Recommendation systems help users find items of interest. Meta-recommendation systems provide users with personalized control over the combination of recommendation data from multiple information sources. In the process, they provide users with more helpful recommendations by allowing users to indicate how important each parameter is in their decision process, and how data should be weighted during recommendation generation. Most current meta-recommendation systems require the submission of large, form-based queries prior to the receipt of recommendations. Such systems make it difficult for a user to conclude what effect a given requirement has on the overall recommendations. This paper considers the construction of an interface to allow dynamic queries for a meta-recommender. It is believed that the addition of a dynamic query interface will provide users with more meaningful meta-recommendations by allowing them to explore these causes and effects.

Keywords
Meta-recommendation systems recommendation system, collaborative filtering, dynamic query interface.

INTRODUCTION
On a daily basis we are faced with information overload as we choose from an overwhelming number of options. To keep abreast of the latest developments in our career field, we can choose from a whole host of journal articles, conference proceedings, textbooks, and web sites. During our personal time we must choose which television show to watch, which movie to see, which CD to play, or which book to read. The number of options from which to choose in each of these categories is often more than we can possibly process.

Workshop: Beyond Personalization 2005
IUI'05, January 9, 2005, San Diego, California, USA
http://www.cs.umn.edu/Research/GroupLens/beyond2005

Recommender Systems have emerged as powerful tools for helping users reduce information overload. Such systems employ a variety of techniques to help users identify items of interest [5]. For example, a recommender system in the domain of movies might suggest that a user go see *Ladder 49* because she requested films classified as "drama" (query fit using information retrieval), because she has previously liked films starring John Travolta (personalization using information filtering), or because people like her have indicated it was a movie they enjoyed (personalization using collaborative filtering). Regardless of technique, these systems attempt to help users identify the items that best fit their needs, their tastes, or even both.

This paper discusses a class of recommendation interface known as *meta-recommendation systems.* These systems present recommendations fused from "recommendation data" from multiple information sources. Meta-recommendations systems encourage users to provide both ephemeral and persistent information requirements. The systems use this data to produce recommendations that blend query-fit with long-term personalization. For example, MetaLens – a real- time meta-recommender in the domain of movies – might recommend Ladder 49 based not only on a combination of the reasons previously mentioned, but also based on the fact that the user indicated she requires a movie showing in her local theater that starts after 9:00 PM. Furthermore, these systems provide a high level of user control over the combination of recommendation data, providing users with more unified and meaningful recommendations. This paper also presents early work in the development of a dynamic query interface for a meta-recommendation system.

RELATED WORK
The earliest "recommender systems" were information filtering and retrieval systems designed to fight information overload in textual domains. Recommender systems that incorporate information retrieval methods are frequently used to satisfy ephemeral information needs from relatively static databases. Conversely, recommender systems that

incorporate information filtering (IF) methods are frequently used to identify items that match relatively stable and specific information needs in domains with a rapid turnover or frequent additions. Although information retrieval and information filtering are considered fundamentally different tasks [1], they are based on similar techniques. This paper will consider both under the singular term "information filtering." Systems which use IF techniques include RE:Agent [2], Ripper [5], NewT [9], and Amalthaea [10].

Collaborative filtering (CF) is an attempt to facilitate the process of "word of mouth." Users provide the system with evaluations of items that may be used to make "recommendations" to other users. The simplest of CF systems provides generalized recommendations by aggregating the evaluations of the community at large. More advanced systems personalize the process by forming an individualized neighborhood for each user consisting of a subset of users whose opinions are highly correlated with those of the original user. Recommender systems based on collaborative filtering include MovieLens [6], Tapestry [7], GroupLens [12], Ringo [17], and PHOAKS [19].

Hybrid Recommender Systems

As researchers have studied different recommender system technologies, many have suggested that no single technology works for all situations. Thus, hybrid systems have been built in an attempt to use the strengths of one technology to offset the weaknesses of another. Burke [3] discusses several different hybridization methods, but points out that most hybrid systems involve the combination of collaborative filtering with either a content-based (IF) or data mining technique.

Tango [4] recommends articles in the domain of an online newspaper. It does so by creating separate recommendations from CF and IF algorithms and merging these using a separate combination filter. The combination filter employed by Tango uses per-user, per-article weights. The calculation of these weights takes into account the degree of confidence each filter has in a particular document's recommendation, as well as error analysis for each filter's past performance for the user in question.

Torres et al. [20] present the results of several experiments involving TechLens. Similar to Tango, TechLens combines both a collaborative filter and a content-based filter to recommend research papers. In both offline and online studies they consider five different algorithms for combining the recommendations from these filters, including sequential algorithms. These techniques take the recommendations from one filter as a seed to the second filter. They conclude that different algorithms should be used for recommending different kinds of papers, although they discovered that sequential algorithms tend to produce poor results under most circumstances.

The SmartPad supermarket product recommender system [8] 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.

Nakamura and Abe [11] describe a system for the automatic recording of programs using a digital video recorder. They implement a set of "specialist" algorithms that use probabilistic estimation to produce recommendations that are both content-based (based on information about previously recorded shows from the electronic program guide) and collaborative (based on the viewing patterns of similar users).

META-RECOMMENDERS

Consider the following scenario. Mary's 8-year-old nephew is visiting for the weekend, and she would like to take him to the movies. Mary has several criteria for the movie that she will select. She would like a comedy or family movie rated no "higher" than PG-13. She would prefer that the movie contain no sex, violence or offensive language, last less than two hours and, if possible, show at a theater in her neighborhood. Finally, she would like to select a movie that she herself might enjoy.

Traditionally, Mary might decide which movie to see by checking the theater listings in the newspaper and asking friends for recommendations. More recently, her quest might include the use of the Internet to access online theater listings and search databases of movie reviews. Additionally, she might be able to obtain personalized, CF-based recommendations from a web site such as MovieLens. Producing her final selection, however, requires a significant amount of manual intervention; Mary must visit each source to gather the data and then decide how to apply this data in making her final decision.

The hybrid systems mentioned in the previous section are a significant step toward solving problems like Mary's. A hybrid movie recommendation system would provide Mary with lists of movies blended from her long-standing collaborative filtering and content-interest profiles. It is likely, however, that such a system would not offer her the ability to provide information that might improve the recommendations produced by the combination algorithm. For example, if given access to the combination algorithm, Mary could indicate that predictions should be biased less towards the British art films she frequently likes and more toward the family movies appropriate for her nephew, or that the movie should be relatively free of offensive language and last less than two hours.

Prior work [15,16] has defined a new form of hybrid system with the level of user control needed to allow for the meaningful blending of recommendations from multiple techniques and sources. These systems, known as meta-recommenders, provide users with personalized control over the generation of a single recommendation list formed from a combination of rich data using multiple information sources and recommendation techniques. Based on the lessons we learned from existing hybrid systems, we built the MetaLens Recommendation Framework (MLRF), a general architecture for the construction of meta-recommenders. Using this framework, we implemented MetaLens, a meta-recommender for the domain of movies. Much like Mary, who makes her final choice by examining several movie data sources, MetaLens uses IF and CF technologies to generate recommendation scores from several Internet film sites.

The user interface for MetaLens centers on two screens. On the preferences screen, users indicate their ephemeral requirements for their movie search. They do this by providing information concerning nineteen features of movies and theaters including genre, MPAA rating, critical reviews, and distance to the theater (Figure 1). For each feature the user may indicate the specific factors he considers important (e.g., "I want to see a film from the ‘comedy’ or ‘family’ genre"), a weight that indicates how important it is that the recommended movie matches these factors (e.g., "It is very important that the movie I see be one of the genres I selected") and a “Display Info?” selection which indicates that data related to the specific feature should be included with the recommendations. As an example, Figure 1 might represent a portion of Mary's requirements for the movie that she views with her nephew.

The meta-recommendation algorithm is based on the extended Boolean information retrieval algorithm proposed by Salton et al [13] as a way to rank partial matches in Boolean queries in the domain of document retrieval. This algorithm is an ideal initial choice for meta-recommenders. In essence, Mary submits a query that says “I want a movie that is a comedy or family movie rated no “higher” than PG-13, containing no sex, violence or bad language, lasting less than two hours and, showing at a theater in my neighborhood.” A traditional Boolean query of these requirements will return only movies matching ALL of these features. Most users, however, will settle for a movie matching a majority of these features.

MetaLens judges overall query fit based on recommendation scores from these multiple data sources. No attempt is made to resolve potential information conflicts. Instead, each piece of data is converted as-is, and the item match scores combined to calculate a query-fit score for each triple. These recommendations are sorted to contain only the highest-rated triple for each movie – each movie is recommended once in conjunction with the theater and show time that best fits the user's requirements – and the final recommendations displayed. Thus, Figure 2 might represent the MetaLens recommendations concerning which movie Mary should take her nephew to see.

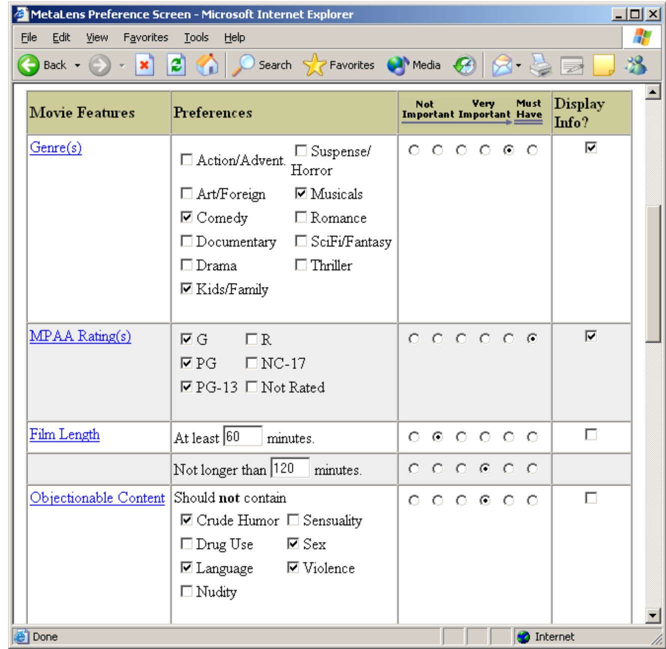
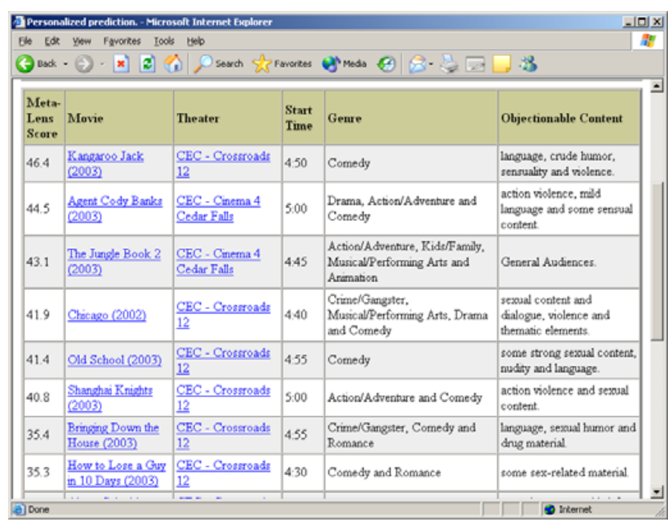


Figure 1: MetaLens Preferences Screen. Users provide information regarding which items among nineteen features are important, the degree to which each recommendations must match the features selected, and what data should be included with final recommendations.



Meta-Lens Score	Movie	Theater	Start Time	Genre	Objectionable Content
46.4	Kangaroo Jack (2003)	CEC - Crossroads 12	4:50	Comedy	language, crude humor, sensuality and violence.
44.5	Agent Cody Banks (2003)	CEC - Cinema 4 Cedar Falls	5:00	Drama, Action/Adventure and Comedy	action violence, mild language and some sensual content.
43.1	The Jungle Book 2 (2003)	CEC - Cinema 4 Cedar Falls	4:45	Action/Adventure, Kids/Family, Musical/Performing Arts and Animation	General Audiences.
41.9	Chicago (2002)	CEC - Crossroads 12	4:40	Crime/Gangster, Musical/Performing Arts, Drama and Comedy	sexual content and dialogue, violence and thematic elements.
41.4	Old School (2003)	CEC - Crossroads 12	4:55	Comedy	some strong sexual content, nudity and language.
40.8	Shanghai Knights (2003)	CEC - Crossroads 12	5:00	Action/Adventure and Comedy	action violence and sexual content.
35.4	Bringing Down the House (2003)	CEC - Crossroads 12	4:55	Crime/Gangster, Comedy and Romance	language, sexual humor and drug material.
35.3	How to Lose a Guy in 10 Days (2003)	CEC - Crossroads 12	4:30	Comedy and Romance	some sex-related material.

Figure 2: MetaLens Recommendation Screen. Users are provided with a list of {movie, theater, showtime} triples ranked according to how well each triple matches the query provided in the preferences screen.

DYNAMIC META-RECOMMENDERS

DynamicLens

One of the advantages of meta-recommenders is that they involve such a rich assortment of recommendation data. Prior research has concluded that users accept meta-recommendation systems such as MetaLens and that they believe such systems provide them with more meaningful recommendations when compared to “traditional” systems [16]. While the current interface is similar to other comparable recommender systems, user interface design experts like Shneiderman [18] would argue that the current interface does not allow users to interact properly with the data. In order for a user of MetaLens to "tweak" a query, the user must return to the preferences screen, modify the requirements or weights for each feature, and resubmit the query to the system. While careful trial and error may indicate the effect different requirements have on the final recommendations, this process can be time consuming and difficult.

This leads to the question how would the recommendation process change as modifications are made to the various interfaces with which users interact. In particular, what would be the affect of “dynamic query” interfaces on the way users interact with a meta-recommender.

In order to begin consideration of answering such a question a new interface was built for MetaLens that employs Shneiderman’s concept of a dynamic query interface. DynamicLens uses the same underlying algorithms and data as MetaLens. However, it merges the preferences interface and the recommendations interface within a single interface. Users select which items are of interest in their query, and to which extent each recommendation should match these items through a preferences panel located on the left side of the interface (Figure 3). Recommendations based on the current query are displayed in the right-hand panel of the interface. Individual changes to the preferences panel generate an automatic and immediate update to the recommendations panel.

There are several potential advantages of a system such as DynamicLens. First, it allows users immediate feedback on the affects of each query requirement. For example, a user adding “violence” to his list of objectionable content can immediately observe which movies drop lower in the recommendations list. Similarly, a user raising the importance factor from “very important” to “Must match” can observe how many movies will be eliminated from contention for not matching her currently selected items. In both cases, the user has the feedback necessary to understand how such requirements affect the final outcome from the recommendation engine. The user can choose to maintain these requirements, select to “soften” the requirement, or choose to eliminate the requirement completely in order to generate recommendations he or she feels will best suit the current needs.

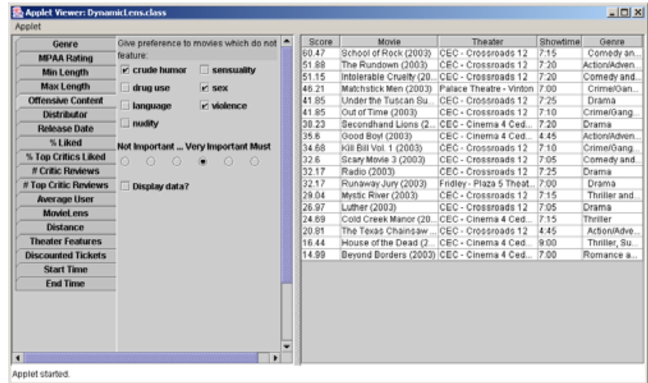


Figure 3: DynamicLens merges the preferences interface with the recommendation interface to provide users with a direct manipulation interface. Users are provided with immediate feedback regarding the effect caused by individual changes to their preferences.

Looking at this from a slightly different angle, a user is better able to interact with a direct query interface in an attempt to discover why a given set of ranking recommendations were made. For example, a user may provide his recommendations and wonder why a given film is recommended so low in the rankings. A dynamic query interface allows him to modify the current recommendation query in an attempt to see which requirement(s) pushed the item so low in the list. In doing so, the user has the ability to gather information that can help him decide that the film really is inappropriate given the current set of requirements.

But is it worth it?

While users were very accepting of MetaLens, it has yet to be shown that they will be similarly accepting of DynamicLens. One advantage of a multi-screen interface like MetaLens is that the interface fits within a standard metaphor of “build query/analyze results.” Furthermore, very few instructions are needed and those that are fit naturally within the appropriate interface.

The potential pitfalls to a dynamic interface like DynamicLens include the decrease in real estate for “natural” instructions, and the increased complexity that comes with adapting what was formerly multiple screens into a single screen. While it is natural in MetaLens to state “tell us what you are looking for” at the top of a long query form, it is less natural to do so when the various elements of the query are divided among multiple tabbed panels. Furthermore, these tabbed panels provide an increase in complexity which may cause a decrease in usefulness – e.g. if a user can’t figure out how to modify the preferences to match his needs, the tool rapidly becomes one with no real use. While the general concept of tabbed option panels has become increasingly common, anecdotal experience has suggested that many users continue to be confused by their use. A review of the research literature failed to yield any usability studies discussing the long-term effectiveness and usability of such an interface. Clearly, DynamicLens will be affected by the overall acceptance of tab-based interfaces.

Acceptance of a new interface is not solely based on a user's perception of usability, however. It is also based on a user's belief in the overall appropriateness of the interface. Several potential benefits of this new interface were proposed in the previous section. However, it remains unclear if users will actually notice that the interface allows them to discern the impact of each recommendation attribute and, thus, detect the benefits of the interface. Worse yet, users may notice, but find the knowledge unimportant. In either case, the complexity introduced into the interface in an effort to provide dynamic queries becomes inappropriate.

It remains to be seen whether controlled user studies will indicate that users consider DynamicLens either usable or appropriate.

CONCLUSIONS

This paper has discussed meta-recommenders – a relatively new way to help users find recommendations that are understandable, usable, and helpful. Furthermore, this paper has considered ways in which the interfaces for such recommenders might be improved through the addition of dynamic queries. All told, it is believed that there is great potential for such interfaces to change the way in which users gather information for decision-making

ACKNOWLEDGMENTS

The author would like to thank the members of the GroupLens Research group at the University of Minnesota who assisted with the early versions of this research.

REFERENCES

1. Belkin, N.J. and Croft, W.B. (1992). Information Filtering and Information Retrieval: Two Sides of the Same Coin? CACM 35(12) pp.29-38.
2. Boone, G. (1998). Concept Features in RE:Agent, an Intelligent Email Agent. Proceedings of Autonomous Agents 98. pp.141-148.
3. Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction, v. 12, pp 331-370.
4. Claypool, M., Gokhale, A., and Miranda, T. (1999). Combining Content-Based and Collaborative Filters in an Online Newspaper. ACM SIGIR Workshop on Recommender Systems.
5. Cohen, W.W. (1996). Learning Rules that classify E-mail. Proceedings of the AAAI Spring Symposium on Machine Learning on Information Access.
6. Dahlen, B.J., Konstan, J.A., Herlocker, J.L., Good, N., Borchers, A., Riedl, J. (1998). Jump-starting movielens: User benefits of starting a collaborative filtering system with "dead data". University of Minnesota TR 98-017.
7. Goldberg, D., Nichols, D., Oki, B.M., and Terry, D. (1992). Using Collaborative Filtering to Weave an Information Tapestry. CACM 35(12) pp.31-70.
8. Lawrence, R.D., Almasi, G.S., Kotlyar, V., Viveros, M.S., and Duri, S.S. (2001). Personalization of Supermarket Product Recommendations. Data Mining and Knowledge Discovery 5(1/2) pp 11-32.
9. Maes, P. (1994). Agents that Reduce Work and Information Overload. CACM 37(7) pp.31-40.
10. Moukas, A. and Zacharia, G. (1997). Evolving a Multi-agent Information Filtering solution in Amalthaea. Proceedings of Autonomous Agents 97 pp.394-403.
11. Nakamura, A. and Abe, N. (2000). Automatic Recording Agent for Digital Video Server. Proceedings of MM-00 pp.57-66.
12. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. CSCW 94, pp. 175-186.
13. Salton, G., Fox, E., and Wu, H. (1983). Extended Boolean Information Retrieval. CACM 26(11) pp.1022-1036.
14. Schafer, J.B., Konstan, J.A., and Riedl, J. (2001). E-Commerce Recommendation Applications. Data Mining and Knowledge Discovery 5(1/2) pp.115-153.
15. Schafer, J.B., Konstan, J.A., and Reidl, J. (2002). Meta-recommendation Systems: User-controlled Integration of Diverse Recommendations. Proceedings of CIKM-02 pp. 196-204.
16. Schafer, J.B., Konstan, J.A., and Reidl, J. (2004). The View through MetaLens: Usage Patterns for a Meta-Recommendation System. IEE Proceedings Software. Anticipated publication December 2004.
17. Shardanand, U. and Maes, P. (1995). Social Information Filtering: Algorithms for Automating Word of Mouth. Proceedings of CHI-95 pp.210-217.
18. Shneiderman, B. and Plaisant, C. (2005). Designing the User Interface: Strategies for Effective Human-Computer Interaction, Fourth Edition. Addison Wesley Longman, Inc.
19. Terveen, L., Hill, W., Amento, B., McDonald, D., and Creter, J. (1997). PHOAKS: A System for Sharing Recommendations. CACM 40(3) pp.59-62.
20. Torres, R., McNee, S., Abel, M., Konstan, J.A., and Riedl, R. (2004). Enhancing Digital Libraries with TechLens. Proceedings of JCDL'04. pp 228-236.