DO YOU HAVE ANY RECOMMENDATIONS?

AN INTRODUCTION TO RECOMMENDER SYSTEMS

PREPARED BY FELICIA POE FOR CDL SERVICES TEAM

LAST UPDATED 07/07/05

I. INTRODUCTION

Increasingly, users expect search engines to deliver *answers* to their queries, no matter how idiosyncratic or imprecise the formulation of their search. No longer content with receiving a list of URLs (often numbering in the thousands) to off-topic sites, users expect that search results be both highly relevant and usefully ranked. Additionally, the growing popularity of collaborative categorization tools, such as the social bookmark manager *del.icio.us* and the photo-sharing site *Flickr,* demonstrates that some users are interested in contributing to the categorization of web content and that this interest extends into the academic community. The ability to automatically narrow, filter, cluster, recommend, and rank search results holds the potential to provide users with quick, accurate answers to their queries.

Recommender systems, typically in the form of either a collaborative filtering system (CF), a content-based filtering system (CBF), or a hybrid of the two (CF-CBF) offer the possibility of recommending to users a targeted set of results ("recommendations") deemed likely to meet the users' need for information. The purpose of this paper is to present an overview of recommender systems, to provide a context in which to analyze how recommender systems might be incorporated into CDL services, and to outline some of the features that contribute to a successful recommender system.

II. RECOMMENDER SYSTEMS: CHARACTERISTICS, STRENGTHS, AND KNOWN PROBLEMS

The term *collaborative filtering* was first introduced in 1992 by a Xerox PARC team experimenting with email filtering systems; early implementations of *content-based filtering* systems are found in information retrieval systems, as well as rule-based expert systems and decision support systems. Currently, there exists a wave of interest in systems that not only *filter* information but also *recommend* alternative items to the user. These filtering systems, whether content-based or collaborative in design, are commonly referred to as *recommender* systems. Filtering methods include:

- **Content-based ("CBF") filtering** – information about the item itself informs the recommendation. Most often utilized in textual domains, recommendations are generated when the content of the item is similar to items the user has liked in the past. Current algorithms are ineffective at analyzing non-textual domains, *e.g.*, audio and film.

    - "One common strategy is to use the text of items a user likes to build a keyword profile and then recommend new items that match the profile. The content-based filtering systems work well when the content of items is amenable to machine processing." (McNee 2002)

    Known problems / shortcomings:

    - "Content-based approaches are based on objective information about the items. This information is automatically extracted from various sources (e.g., Web pages) or manually introduced (e.g., product database). Subjective attributes… are not taken into account." (Montaner 2003)

- "Content-based filtering techniques have no inherent method for generating serendipitous finds. The system recommends more of what the user has already seen and indicated a liking for." (Montaner 2003)

Advantages:

- Previously unrated items may be recommended by extracting information about the content of item (via information extraction, machine-learning algorithm for text categorization, etc.)
- Recommendations can be session-specific, meaning previous selections/ purchases do not influence recommendations within current session. Method can accommodate changing user needs.

- **Collaborative ("CF") filtering** - information drawn from user preferences/ratings inform the recommendation. A type of "social filtering."

  - "CF works by recommending items to people based on what other similar people have previously liked. CF creates neighborhoods of 'similar' users (neighbors) for each user in the system and recommends an item to one user if her neighbors have rated it highly." (Torres 2004)
  - "The system maintains a database of the preferences of individual users, finds other users whose known preferences correlate significantly with a given patron, and recommends to a person other items enjoyed by his or her matched patrons." (Mooney 2000)

  Known problems / shortcomings:

  - "First-rater problem: items need to be rated by at least one neighbor to be recommended, so the item cannot be recommended until someone rates it first." (Torres 2004)
  - "Sparsity problem: in many domains, a user is likely to rate only a very small percentage of the available items. This can make it difficult to find agreement among individuals, since they may have little overlap in the set of items they've rated." (Torres 2004)
  - "When a CF system is first created, there are many items in the system, few users in the system, and no ratings. Without ratings, the system cannot generate recommendations and users see no benefit. Without users, there is no way for new ratings to be entered into the systems. When applying CF to a domain, it is valuable to seek preexisting data that can be used to seed such a database of ratings." (McNee 2002)
  - Regarding the problems with using collaborative filtering within the OPAC environment (Mooney 2000):
    - Majority of library books are utilized by very few patrons.

        - CF techniques tend to recommend popular titles, perpetuating homogeneity in reading choices.
        - CF requires information about reading habits, raising privacy concerns.
        - Although a new item may be of interest to a user, CF techniques are unable to recommend items that have yet to be circulated, rated, etc.

- **Hybrid systems (CF-CBF)** – combination of collaborative filtering (CF) and content-based filtering (CBF) systems. Attempts to use the strengths of each to compensate for the inherent weakness of the other. Two popular hybrid approaches include developing an algorithm that is composed of two separate (CF and CBF) modules. In the first approach, the algorithms are sequentially applied to the dataset; in the second approach, results are merged before system makes its final recommendation.

    "In many ways, collaborative and content-based approaches provide complimentary capabilities. Collaborative methods are best at recommending reasonable well-known items to users in communities of similar tastes when sufficient user data is available but effective content information is not. Content-based methods are best at recommending unpopular items to users with unique tastes when sufficient other data is unavailable but effective content information is easy to obtain." (Mooney 2000)

    Note: See "Enhancing Digital Libraries with TechLens" (Torres 2004) for an excellent example of a CF-CBF hybrid algorithm applied to a repository of research papers.

    ***Examples**:* Amazon employs a CF-CBF hybrid technique; MovieLens employs a CF-CBF hybrid technique.

- **Characteristics of recommender systems:**
  Note: Below quoted directly from Zhu (2005)

    - All systems require a model of the user's interests, but some learn the model and some do not.
    - Some systems require a training phase in which users distinguish content they desire from content they do not.
    - Systems vary in the extent to which they can use information learned from specific users (individual) and groups of users (group or population).
    - Systems vary according to how they validate the recommendations they make. Some use indirect information contained in correlations whereas others use explicit direct judgments of content.
    - Some systems take the sequence of pages into account, and some do not.

III. EVALUATING RECOMMENDER SYSTEMS: THE USER PERSPECTIVE

Jonathan Herlocker of Oregon State University is a leader in researching the design and evaluation of recommender systems. Herlocker's approach is user-centric, and he suggests a primary (and difficult) early design task is deciding what key user tasks the recommender system will be designed to support.

- **Define user tasks**: Clearly identifying what user tasks the recommender system will support is essential to good system design. Only after the primary user tasks have been identified can functional specifications be defined, an algorithm to support defined tasks be developed, and a dataset for testing the algorithm be selected. Herlocker suggests user tasks typically fall into the following categories:

- Find some good items – discover "best bets", novel items, item ranking, etc.
- Find all good items – recall more important than accuracy.
- Recommend sequence – discover which item to read first, which to read next, which items are for beginners, which are for experts.
- Just browsing – user browsing to discover, to learn, to be entertained.
- Improve profile – user contributes ratings in effort to improve recommendations received.
- Express self – fulfill a desire to participate in a forum, to let opinion be known.
- Help others – user with particular domain knowledge contributes ratings in effort to educate others with limited expertise.
- Influence others – user attempt to influence recommendations by voting/ranking repeatedly; common in movie ratings, book ratings, etc.

- **Design an accurate, useful, usable system**: Most research into recommender systems focus on issues of accuracy, algorithms, and system software design. User-focused research is somewhat limited and more likely to be presented at conferences versus published in peer-reviewed journals. Characteristics of a useful and usable recommender system include:

    - Results should be both accurate and useful. Accuracy is easier to accomplish as compared to usefulness. A system may accurately recommend easy-to-predict ("low hanging fruit") items, but those items might only be useful to users possessing limited knowledge of a subject. Challenge: users who frequently shift focus, explore unfamiliar features, or wander through system in off-topic fashion.

    - Results should be *suitable*. Herlocker defines suitability in terms of coverage, learning rate, novelty and serendipity, and confidence. (Herlocker 2004)

        - *Coverage* pertains to the number of items within the entire system for which items can be recommended. Limited coverage sometimes referred to as the "sparsity" problem
        - *Learning rate* refers to the number of data points necessary for an algorithm to begin returning acceptable results. Slow learning rates, or situations where limited ratings are available to inform recommendations are referred to as the "first rater" problem.
        - *Novelty and serendipity* refers to the ability of a system to broaden the user's interests over time. Recommendations that are obvious (e.g., "if you like Hamlet you might also want to read King Lear") are not necessarily useful. Although obvious recommendations may be valued by the novice, recommendations should also contain elements of novelty and serendipity.
        - *Confidence* pertains to a system's ability to rate its strength of recommendation.

- **Design a system and UI that is transparent - explain to the user w*hy* a recommendation is being suggested.** Users approach recommender systems with a mental model of how recommendations are determined by the system. These mental models are often misinformed and may lead to user frustration. For example, a user may possess the mental model that book recommendations are based on items sharing similar *content-based* characteristics, such as the same author or similar subject headings. If the system, however, is a collaborative filtering system that bases recommendations on previously stated user preferences (versus item characteristics) then the user may loose trust in the system when it fails to recommend items that seem obvious. Users who understand *why* an item is being recommended report a higher degree of confidence, liking, and understanding. (Sinha 2002)

The system and user interface should be designed to indicate why a recommendation is being made. Examples include:

- "Readers of this book have also read … " (collaborative filtering)
- "Other books by this author include … " (content-based filtering)
- "People who purchased this book also purchased …" (collaborative filtering)
- "This book is similar to other books you rated highly … " (collaborative filtering)

Herlocker (2000) suggests that moving the recommendation process from the "black box" to the "white box" model provides transparency and improves user trust and acceptance of system. Benefits include:

- *Justification* – User understands why a recommendation is being made and can decide how much confidence to place in the recommendation.
- *User involvement* – User applies her own knowledge and inference skills to the recommendation process.
- *Education* – User is educated as to system strengths and weakness.

- **Provide users with information about the item being recommended.** Users report a higher satisfaction with recommendations that contain basic information about the item being recommended. For example, a book recommendation should contain author, title, year of publication, and possibly a thumbprint of the jacket cover or a brief description. Users expressed a desire to be able to peruse a list of recommendations and to have their memory "jogged" as to whether they are familiar with the item. They did not want to have to click through in order to ascertain a recommendation's usefulness. (Swearingen & Sinha 2000)

- **User perception of usefulness is influenced by factors pertaining to both the system algorithm and the user interface.** Although the ability to deliver good recommendations is primarily a factor of the accuracy of the system algorithm, UI features also contribute to user satisfaction. Factors relating to the algorithm include: the number of good recommendations, the number of useful recommendations, and a proper balance of obvious and novel recommendations. Factors pertaining to the user interface include: the level of detail in item descriptions, system transparency, and the ability to refine recommendations by date, genre, etc. (Swearingen & Sinha 2000)

IV. PROFILE GENERATION AND MAINTENANCE

Note: Below drawn directly from Montaner (2003)

The generation and maintenance of accurate user profiles is an essential component of a successful recommender system. Determining similar user's interests, and reflecting those interests back in the form of appropriate recommendations, are primary functions of a recommender system. User profile generation and maintenance require five primary design decisions (Montaner 2003):

- **Profile representation technique**: By what action(s) will profiles be tracked?

- History-based model: navigation history, purchase history, etc.
- Vector space model: represent each item as a vector in a vector space, allowing items with similar content to be assigned similar vectors.
- Weighted n-grams: based on word structure and character occurrence.
- Weighted semantic networks: based on meanings of words -> creation of networks -> connection of networks to users' interests.
- Weighted associative networks: based on terms and concepts in which user is interested.
- Classifier-based models: based on user profiling learning technique; utilizes training sets.
- User-item ratings matrix: based on historical user ratings of items; does not utilize training sets.
- Demographic features: based on stereotype representing user's demographic features.

***Examples**:* Amazon employs a purchase history with ratings technique; MovieLens employs a weighted feature vector technique.

- **Technique used to generate the initial profile**: Complex aspect of system design, as users typically do not want to expend effort on defining interests or establishing a profile. (Montaner 2003)

  - Empty: profile built through recognition of interactions (history-based model).
  - Manual: user required to list/register interests.
  - Stereotyping: user required to complete form containing demographic data.
  - Training set: user required to rate examples indicating interest, e.g., relevant/ irrelevant.

  Examples: Amazon employs an empty technique; MovieLens employs a training set technique.

- **Profile learning technique**: System builds profiles via interaction with above information. (Montaner 2003)

  - Not necessary: system has already acquired information from user registration process.
  - Structured information retrieval technique: typically, term-frequency/inverse-document frequency (TF-IDF).
  - Clustering: Similar users are grouped; system assumes members of a group share interests.
  - Classifiers: Automated classification techniques employing machine-learning strategies.

  ***Examples**:* Amazon's system does not require profile learning (not necessary); MovieLens employs a structured information retrieval (TF-IDF) technique.

- **Relevance feedback technique**: Technique by which the system receives and updates user's profile. Typically based on positive information (items selected or purchased by user) or negative information (not selecting or purchasing an item infers non-interest). (Montaner 2003)

- No feedback: system does not automatically update a profile, so no relevance feedback is required. If desired, user must manually update profile.
- Explicit feedback: typically utilized in systems that require users to indicate like/dislike, participate in ratings, or provide text feedback. Advantage: simple system design; disadvantages: user reluctance to participate in requests for feedback. (Pazzani reports participation level of 15%.)
- Implicit feedback: system infers preferences by monitoring user's actions, including links followed, click paths, purchase history, navigation history, time spent on a web page, and processing actions such as saving/printing/deleting a document, creating a bookmark, scrolling/maximizing/resizing a window. Also referred to as "behavior based". (See: Behavior-based Recommender Systems for Web Content by Tingshao Zhu)
- Hybrid approach: combination of explicit and implicit feedback techniques.

  ***Examples**:* Amazon employs a hybrid approach of explicit (ratings) and implicit (purchase history); MovieLens employs an explicit (ratings) technique.

- **Profile adaptation technique**: The ability for the system to adapt and reflect new user interests and disregard outdated ones. Essential design element, as users' interests change over time. (Montaner 2003)

  - Manual: user required to update list of interests.
  - Add new information: information added based on relevance feedback technique; disadvantages include inability to delete outdated interests.
  - Gradual forgetting function: recent user relevance feedback is positively weighted, resulting in a gradual "forgetting" of earlier interactions.

    ***Examples**:* Amazon employs an "add new information" technique; MovieLens employs an "add new information" technique.

- **User profile matching** – Systems employing collaborative filtering (including CF-CBF hybrid systems) typically determine matches via a process of identifying similar users -> creating a neighborhood of users -> determining recommendations based on selected neighbors. (Montaner 2003)

  - Find similar users – technique employing standard similarity measures, including:
    - Nearest neighbor
    - Clustering
    - Classification
  - Create a neighborhood – techniques used include the creation of centroids, correlation-thresholding, and best-n-neighbors.
  - Computing a prediction based on selecting neighbors – techniques include:
    - Most-frequent item recommendation
    - Association rule-based recommendation
    - Weighted average of ratings

V: Quality assurance, algorithms, datasets

Testing the system algorithm, and the choice of test dataset, is a crucial component of successful system design. The metric by which an algorithm is rated should reflect the user task the system has been designed to support. (Herlocker 2004)

- **On what dataset will the filtering algorithm be tested?** The test dataset should possess properties that lend themselves to accurate user-task modeling. Possibilities include: live user experiments, offline analysis, natural datasets, and synthesized datasets.

  Herlocker suggests that dataset properties can be broken into three categories: domain features, inherent features, and sample features. I list the features below as they provide valuable insight into how functional specifications might be approached.
  Note: Three sections below drawn directly from Herlocker (2004)

  - "**Domain features** reflect the nature of the content being recommended, rather than any particular system." Possible domain features include:

    - the content topic being recommended/rated and the associated context in which the rating/recommendation takes place;
    - the user tasks supported by the recommender;
    - the novelty need and the quality need;
    - the cost/benefit ratio of false/true positives/negatives;
    - the granularity of the user preferences.

  - "**Inherent features** reflect the nature of the specific recommender system from which the data was drawn (and possibly from its collection practices)." Possible inherent features include:

    - whether ratings are explicit, implicit, or both;
    - the scale on which items are rated;
    - the dimensions of rating;
    - the presence or absence of a timestamp on ratings;
    - whether the recommendations displayed to the user were recorded.

  - "**Sample features** reflect distribution properties of the data, and often can be manipulated by selecting the appropriate subject of a larger dataset."

REFERENCES

LAST UPDATED: 05/24/05

Bilgic, Mustafa
Explaining recommendations: satisfaction vs. promotion
Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research
IUI’05, January 9, 2005, San Diego, California, USA
http://www.cs.umn.edu/Research/GroupLens/beyond2005/
http://www.cs.umn.edu/Research/GroupLens/beyond2005/full/bilgic.pdf

Cosley, Dan
Is seeing believing?: how recommender system interfaces affect users' opinions
Proceedings of the SIGCHI conference on Human factors in computing systems
Ft. Lauderdale, Florida, USA
Pages: 585 – 592 (2003)
http://doi.acm.org/10.1145/642611.642713

Decoste, Dennis
Recommender systems research at Yahoo! research labs
Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research
IUI’05, January 9, 2005, San Diego, California, USA
http://www.cs.umn.edu/Research/GroupLens/beyond2005/
http://www.cs.umn.edu/Research/GroupLens/beyond2005/position/decoste.pdf

Herlocker, Jonathan L.
Evaluating collaborative filtering recommender systems
ACM Transactions on Information Systems (TOIS)
Volume 22 , Issue 1 (2004)
Pages: 5 - 53
http://doi.acm.org/10.1145/963770.963772

Herlocker, Jonathan L.
Explaining collaborative filtering recommendations
Proceedings of the 2000 ACM conference on Computer supported cooperative work
Philadelphia, Pennsylvania, United States
Pages: 241 – 250 (2000)
http://doi.acm.org/10.1145/358916.358995

Kruk, Sebastian Ryszard
FOAFRealm: Making Social Collaborative Filtering Real
Website: MarcOnt Initiative: Bibliographic description and related tools utilising Semantic Web technologies
http://www.marcont.org/pages/eswc2005_foafrealm.pdf

McNee, Sean M.
Interfaces for eliciting new user preferences in recommender systems
Proceedings of the 9th International Conference on User Modeling
Minneapolis, MN, USA
LNAI 2702
Pages: 178-188 (June 2003)
http://www-users.cs.umn.edu/~mcnee/mcnee-um2003.pdf

McNee, Sean M.
On the recommending of citations for research papers
Computer Supported Cooperative Work
Proceedings of the 2002 ACM conference on Computer supported cooperative work
New Orleans, Louisiana, USA
Pages: 116 – 125 (2002)
http://doi.acm.org/10.1145/587078.587096

Montaner, Miquel
A Taxonomy of Recommender Agents on the Internet
Artificial Intelligence Review
Volume 19, Issue 4 (June 2003)
Pages: 285 - 330
http://springerlink.metapress.com/openurl.asp?genre=article&id=doi:10.1023/A:1022850703159

Mooney, Raymond J.
Content-based book recommending using learning for text categorization
International Conference on Digital Libraries
Proceedings of the fifth ACM conference on Digital libraries
San Antonio, Texas, USA
Pages: 195 – 204 (2000)
http://doi.acm.org/10.1145/336597.336662

Schafer, J. Ben
DynamicLens: a dynamic user-interface for a meta-recommendation system
Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research
IUI'05, January 9, 2005, San Diego, California, USA
http://www.cs.umn.edu/Research/GroupLens/beyond2005/
http://www.cs.umn.edu/Research/GroupLens/beyond2005/full/schafer.pdf

Sinha, Rashmi
The role of transparency in recommender systems
Conference on Human Factors in Computing Systems
CHI '02 extended abstracts on Human factors in computing systems
Minneapolis, Minnesota, USA
Pages: 830 – 831 (2002)
http://doi.acm.org/10.1145/506443.506619

Swearingen, K.
Beyond algorithms: An HCI perspective on recommender systems
Proceedings of the SIGIR 2001 Workshop on Recommender Systems
2001
http://www.rashmisinha.com/articles/BeyondAlgorithms.html

Torres, Roberto
Enhancing digital libraries with TechLens+
International Conference on Digital Libraries
Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries
Tuscon, AZ, USA
Pages: 228 – 236 (2004)
http://doi.acm.org/10.1145/996350.996402

Zhang, Xiangmin
Issues of applying collaborative filtering recommendations in information retrieval
Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research
IUI'05, January 9, 2005, San Diego, California, USA
http://www.cs.umn.edu/Research/GroupLens/beyond2005/
http://www.cs.umn.edu/Research/GroupLens/beyond2005/position/zhang.pdf

Zhu, Tingshao
Behavior-based recommender systems for Web content
Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research
IUI'05, January 9, 2005, San Diego, California, USA
http://www.cs.umn.edu/Research/GroupLens/beyond2005/
http://www.cs.umn.edu/Research/GroupLens/beyond2005/full/zhu.pdf

PROFESSORS, PROJECTS, DATASETS, OPEN SOURCE SOFTWARE

Carnegie Mellon Data Privacy Lab
References to privacy-preserving data mining literature
(Includes papers on privacy preserving collaborative filtering.)
http://privacy.cs.cmu.edu/dataprivacy/papers/ppdm/

CoFE 0.4 (open source)
http://eecs.oregonstate.edu/iis/CoFE/

"The Intelligent Information Systems (IIS) research group of Oregon State University is pleased to announce Version 0.4 of CoFE: a free Java recommendation engine for collaborative filtering. CoFE is short for 'COllaborative Filtering Engine'.

This free server (including source code) allows anyone to easily set up a recommendation system. It should run on any platform that supports Java 1.4. We hope it will enhance academic research and commercial application of such systems by removing the need for developers to create such an engine themselves."

GroupLens Research Project
http://www.cs.umn.edu/research/GroupLens/index.html

A research group in the Department of Computer Science and Engineering at the University of Minnesota. Members of the GroupLens Research Project are involved in many research projects related to the fields of information filtering, collaborative filtering, and recommender systems. The project is lead by professors John Riedl and Joseph Konstan. Includes links to following datasets:

BookCrossing (BX) dataset
Jester Joke Recommender System dataset
MovieLens datasets

Herlocker, Jonathan L.
Oregon State University School of EECS
Website: Research papers by Jonathan Herlocker
http://eecs.oregonstate.edu/iis/CoFE//?q=taxonomy_menu/9/9

LIBRA: Learning Intelligent Book Recommending Agent
Project from Raymond Mooney (UTA) using machine learning to recommend books. Utilizes database of information about books extracted from Amazon.com.
http://www.cs.utexas.edu/users/mooney/libra/

McNee, Sean M.
University of Minnesota Department of Computer Science and Engineering
Personal website
http://www-users.cs.umn.edu/~mcnee/

Mooney, Raymond J.
University of Texas, Department of Computer Sciences
Website: Learning for Recommender Systems
http://www.cs.utexas.edu/users/ml/publication/recommender.html

Yahoo! Research Labs
Spot Workshop on Recommender Systems
Pasadena, CA
August 26, 2004
http://research.yahoo.com/~pennockd/spot/rs/