Workshop Chairs

Ido Guy, IBM Haifa Research Lab, Israel
Li Chen, Hong Kong Baptist University, Hong Kong
Michelle X. Zhou, IBM China Research Lab, China

Program Committee

Mathias Bauer, Mineway, Germany
Shlomo Berkovsky, CSIRO, Australia
Peter Brusilovsky, University of Pittsburgh, USA
Robin Burke, DePaul University, USA
David Carmel, IBM Research, Israel
Alexander Felfernig, Graz University of Technology, Austria
Werner Geyer, IBM Research, USA
Alfred Kobsa, University of California, USA
Yehuda Koren, Yahoo, Israel
John Riedl, University of Minnesota, USA
Bracha Shapira, Ben-Gurion University of the Negev, Israel
Foreword

Social media sites have become tremendously popular in recent years. These sites include photo and video sharing sites such as Flickr and YouTube, blog and wiki systems such as Blogger and Wikipedia, social tagging sites such as Delicious, social network sites (SNSs), such as MySpace and Facebook, and micro-blogging sites such as Twitter. Millions of users are active daily in these sites, creating rich information online that has not been available before. Yet, the abundance and popularity of social media sites floods users with huge volumes of information and hence poses a great challenge in terms of information overload. In addition, most of user-generated contents are unstructured (e.g., blogs and wikis). It hence raises open questions of how such information can be exploited for personalization.

Social Recommender Systems (SRSs) aim to alleviate information overload over social media users by presenting the most attractive and relevant content, often using personalization techniques adapted for the specific user.

SRSs also aim at increasing adoption, engagement, and participation of new and existing users of social media sites. Traditional techniques, such as content-based methods and collaborative filtering are being used separately and jointly to support effective recommendations. Yet, the social media platform allows incorporating new techniques that take advantage of the new information becoming publicly available in social media sites, such as the explicit connections between individuals in SNSs, the tags people are using to classify items, and the content they create. In addition to recommending content to consume, new types of recommendations emerge within social media, such as of people and communities to connect to, to follow, or to join.

The fact that much of the information within social media sites – tags, comments, ratings, connections, content, and sometimes even message correspondence – is public, enabling more transparency in social recommender systems. New techniques for explanations that try to reason a recommendation provided to a user are being exploited, aiming at increasing users’ trust in the system and stimulating more active participation. On the other hand, incorporating user feedback – both explicit and implicit – to improve recommendations and keep them attractive over time is another important challenge for SRSs.

Indeed, explaining the rationale behind recommendations as well as presenting recommendation results is an important aspect of social recommender systems. Because of the diverse information used in making recommendation (e.g., social network as well as content relevance), effective mechanisms must be in place to explain the recommendation rationale and results to users. Not only will such an explanation help instil users’ trust in recommended items, but it also provides an opportunity for users to provide feedback for adaptive recommendations (e.g., deleting unwanted information sources to be used for making recommendations). In addition to providing recommendations to individuals, social recommender systems are also often targeted for communities. Community recommendations need to take into account the entire set of community members, the aggregation of their diverse needs for constructing community preference models, the analysis of their collective behavior, and the different
content already consumed by the community.

Another main challenge in the area of Recommender Systems is the evaluation of provided recommendations. Social media presents opportunities for new evaluation techniques, for example by leveraging tags as interest indicators of specific topics or specific items, or by harnessing the crowds that are actively participating in the sites. Developing new evaluation techniques and applying them on social recommender systems are essential to compare different recommendation methods and reach more effective systems.

This workshop brought together researchers and practitioners around the emerging topic of recommender systems within social media in order to: (1) share research and techniques used to develop effective social media recommenders, from algorithms, through user interfaces, to evaluation (2) identify next key challenges in the area, and (3) identify new cross-topic collaboration opportunities.

**The Workshop Organizing Committee**

February 2010
Table of Contents

1. Information Seeking with Social Signals: Anatomy of a Social Tag-based Exploratory Search Browser, Ed H. Chi and Rowan Nairn
2. Tags Meet Ratings: Improving Collaborative Filtering with Tag-Based Neighborhood Method, Zhe Wang, Yongji Wang, Hu Wu
3. Ethical Intelligence in Social Recommender Systems, Peter Juel Henrichsen
5. User Evaluation Framework of Recommender Systems, Li Chen, Pearl Pu
6. A Step toward Personalized Social Geotagging, A. Di Napoli, F. Gasparetti, A. Micarelli, G. Sansonetti
7. Relationship Aggregation for Enhancing Enterprise Recommendations, Ido Guy
8. Pharos: Social Map-Based Recommendation for Content-Centric Social Websites, Wentao Zheng, Michelle Zhou, Shiwan Zhao, Quan Yuan, Xiatian Zhang, Changyan Chi
9. Conference Navigator 2.0: Community-Based Recommendation for Academic Conferences, Chirayu Wongchokprasitti, Peter Brusilovsky, Denis Parra
Information Seeking with Social Signals: Anatomy of a Social Tag-based Exploratory Search Browser

Ed H. Chi, Rowan Nairn
Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304 USA
{echi,rnairn}@parc.com

ABSTRACT
Whereas for the fact-retrieval searches, optimal paths to the documents containing the required information are crucial, learning and investigation activities lead to a more continuous and exploratory process with the knowledge acquired during this “journey” being essential as well. Therefore, information seeking systems should focus on providing cues that might make these explorations more efficient. One possible solution is in building information seeking systems in which navigation signposts are provided by social cues provided by a large number of other people. One possible source for social cues is all of the social bookmarks on social tagging sites. Social tagging arose out of the need to organize found information that is worth revisiting. The collective behavior of users who tagged contents seems to offer a good basis for exploratory search interfaces, even for users who are not using social bookmarking sites. In this paper, we present the algorithm of a tag-based exploratory system based on this idea.

Author Keywords
Social Tagging, Exploratory Interfaces, Social Search

ACM Classification Keywords
H3.3 [Information Search and Retrieval]: Relevance Feedback, Search Process, Selection Process; H5.2. [Information interfaces and presentation]: User Interfaces

INTRODUCTION
Existing search engines on the Web are often best on search tasks that involve finding a specific answer to a specific question. However, users of web search engines often need to explore new topic areas, specifically looking for general coverage of a topic area to provide an overview. As part of information seeking, these kinds of exploratory searches involve ill-structured problems and more open-ended goals, with persistent, opportunistic, iterative, multi-faceted processes aimed more at learning than answering a specific query [18, 23].

One existing solution to exploratory search problems is the use of intelligent clustering algorithms that groups and organizes search results into broad categories for easy browsing. Clusty from Vivisimo (clusty.com) is one relatively successful example of these kinds of systems that grew out of AI research at Carnegie Mellon University. One well-known example is Scatter/Gather, which used fast clustering algorithm to provide a browsing interface to very large document collections [6]. These efforts can be seen as continuing a long line of search system research on user relevance feedback, which is a set of techniques for users to have an interactive dialog with the search system, often to explore a topic space [2, 21]. These clustering-based browsing systems extract patterns in the content to provide grouping structures and cues for users to follow in their exploratory searches, often narrowing down on more specific topic areas or jumping between related sub-topics.

Several researchers have suggested the possibility of aggregating social cues from social bookmarks to provide cues in social search systems [13]. We wish to seriously explore the use of social cues to provide navigational aids to exploratory users. The problem with freeform social tagging systems is that, as the tagging systems evolve over time, their information signal declines and noise increases, due to synonyms, misspellings, and other linguistic morphologies [4].

We designed and implemented a tag-based exploratory search system called MrTaggy.com, which is constructed with social tagging data, crawled from social bookmarking sites on the web. Based on the TagSearch algorithm, MrTaggy performs tag normalizations that reduces the noise and finds the patterns of co-occurrence between tags to offer recommendations of related tags and contents [15]. We surmised that the related tags help deal with the vocabulary problem during search [10]. The hope is that these recommendations of tags and destination pages offer support to the user while exploring an unfamiliar topic area.

In a recent paper [15], we described a user experiment in which we studied the learning effects of subjects using our tag search browser as compared against a baseline system.
We found that MrTaggy’s full exploratory features provide to the users a kind of scaffolding support for learning topic domains, particularly compensating for the lack of prior knowledge in the topic area.

However, due to space limitations, the previous paper did not present any details on the implementation and algorithm design of the system. In this paper, we detail the design and implementation of MrTaggy.com.

First, we briefly give an overview of the overall user interface and system. Then we focus specifically on a deeper discussion of the design choices we made in the system, as well as the MapReduce architecture needed to model and process 140 million bookmarks using a probabilistic graph model. We then provide a quick overview of the user study reported previously, and finally offer some concluding remarks.

RELATED WORK
Much speculation in the Web and the search engine research community has focused on the promise of “Social Search”. Researchers and practitioners now use the term “social search” to describe search systems in which social interactions or information from social sources are engaged in some way [7]. Current social search systems can be categorized into two general classes:

1) Social answering systems utilize people with expertise or opinions to answer particular questions in a domain. Answerers could come from various levels of social proximity, including close friends and coworkers as well as the greater public. Yahoo! Answers (answers.yahoo.com) is one example of such systems. Early academic research includes Ackerman’s Answer Garden [1], and recent startups include Aardvark (vark.com) and Quora (quora.com).

Some systems utilize social networks to find friends or friends of friends to provide answers. Web users also use discussion forums, IM chat systems, or their favorite social networking systems like Facebook and Twitter to ask their social network for answers that are hard to find using traditional keyword-based systems. These systems differ in terms of their immediacy, size of the network, as well as support for expert finding.

Importantly, the effectiveness of these systems depends on the efficiency in which they utilize search and recommendation algorithms to return the most relevant past answers, allowing for better constructions of the knowledge base.

2) Social feedback systems utilize social attention data to rank search results or information items. Feedback from users could be obtained either implicitly or explicitly. For example, social attention data could come from usage logs implicitly, or systems could explicitly ask for votes, tags, and bookmarks.

Many researchers in the information retrieval community have already explored the use of query logs for aiding later searchers [20, 8, 11]. Direct Hit1 was one early example from early 2001 that used click data on search results to inform search ranking. The click data was gathered implicitly through the search engine usage log.

Others like Google’s SearchWiki are allowing users to explicitly vote for search results to directly influence the search rankings. Indeed, vote-based systems are becoming more popular recently. Interestingly, Google’s original ranking algorithm PageRank could also be classified as an implicit voting system by essentially treating a hyperlink as a vote for the linked content.

Popularity data derived from other social cues could also be used in ranking search results. Several researchers in CSCW have noted how bookmarks and tags serve as signals to others in the community. For example, Lee found that analyses of del.icio.us users who perceive greater degrees of social presence are more likely to annotate their bookmarks to facilitate sharing and discovery [17]. A well-known study by Golder and Huberman showed that there is remarkable regularity in the structure of the social tagging systems that is suggestive of a productive peer-to-peer knowledge system [12].

In both classes of social search systems, there are still many opportunities to apply sophisticated statistical and structure-based analytics to improve search experience for social searchers. For example, expertise-finding algorithms could be applied to help find answerers who can provide higher-quality answers to particular questions in social answering systems. Common patterns between question-and-answer pairs could be exploited to construct semantic relationships, which may be used to draw automatic inferences to new questions. Data mining algorithms could construct ontologies that are useful for browsing through the tags and bookmarked documents.

SOCIAL BOOKMARKS AS NAVIGATION SIGNPOSTS
Recently there has been an efflorescence of systems aimed at supporting social information foraging and sensemaking. These include social tagging and bookmarking systems for photos (e.g., flickr.com), videos (e.g., youtube.com), or Web pages (e.g., del.icio.us). A unique aspect of tagging systems is the freedom that users have in choosing the vocabulary used to tag objects: any free-form keyword is allowed as a tag.

Tagging systems provide ways for users to generate labeled links to content that, at a later time, can be browsed and searched. Social bookmarking systems such as del.icio.us already allow users to search the entire database for websites that match particular popular tags. Tags can be

---

1 http://www.searchengineshowdown.com/features/directhit/review.html
organized to provide meaningful navigation structures, and, consequently, can be viewed as an external representation of what the users learned from a page and of how they chose to organize that knowledge.

Using social tagging data as “navigational advice” and suggestions for additional vocabulary terms, we are interested in designing exploratory search systems that could help novice users gain knowledge in a topic area more quickly.

However, one problem is that the social cues given by people are inherently noisy. Social tagging generates vast amounts of noise in various forms, including synonyms, misspellings, and other linguistic morphologies, as well as deliberate spam [4]. In past research, we showed that extracting patterns within such data becomes more and more difficult as the data size grows [4]. This research shows that an information theoretic analysis of tag usage in del.icio.us bookmarks suggest of decreased efficiency in using tags as navigational aids [4].

To combat noisy patterns in tags, we have designed a system using probabilistic networks to model relationships between tags, which are treated as topic keywords. The system enables users to quickly give relevance feedbacks to the system to narrow down to related concepts and relevant URLs. The idea here is to bootstrap the user quickly with other related concepts that might be gleaned from social usage of related tags. Also, the popularities of various URLs are suggestive of the best information sources to consult, which the user can use as navigational signposts.

THE TAGSEARCH ALGORITHM

Here we describe an algorithm called TagSearch that uses the relationships between tags and documents to suggest other tags and documents. Conceptually, given a particular tag, for tag suggestion, we want to construct a semantic similarity graph as in Figure 1.

![Semantic Similarity Graph](image)

**Figure 1:** Conceptual Semantic Similarity Relationships between Tags.

Of course, for a URL, if we want to suggest other URLs, we also want to form a similar semantic similarity graph, which is mediated by the tags. There are also the cases where, given a tag, we want to suggest URLs, and vice versa.

In our approach, the idea is to first form a bigraph between document and tagging pairs. Each tagging data in a tuple specifies a linking relationship between a tag and a document object. For each URL, we want to know the probability of a particular tag being relevant to that URL, and vice versa. For a URL, the probability $p(\text{Tag|URL})$ can be roughly estimated by the number of times a particular tag is applied by users divided by total number of times all tags are used for a URL. Figure 2 depicts this bi-graph.

![bigraph between document/tag](image)

**Figure 2:** bigraph between document/tag.

For a tag, we want to also compute the probability that a URL is related to it. This probability, $p(\text{URL|Tag})$, can be estimated by dividing the number of times an URL is tagged with a particular tag divided by the total number of times the URL is tagged.

A sketch of the idea behind the algorithm is as follows.

**To suggest tags:**

1. What we want to do is then form a “tag profile” for a tag, which is the set of other tags that are related to the tag. To compute the tag profiles, we use the bigraph to perform a spreading activation to find a pattern of other tags that are related to a set of tags. Once we have the tag profiles, we can find other tags that are related by comparing these tag profiles. That is, for a given tag, we can compare its tag profile to other tag profiles in the system to find the top most related tags.

2. Another way to do the same thing is to form a “document profile” for a tag, which is the set of other documents that are related to the tag, similarly using spreading activation. We can then find other tags that are related using these document profiles.

**To suggest documents:**

3. We can form “tag profiles” for a document, which is the set of other tags that are related to that document, again using the spreading activation method. We can then
compare these tag profiles for documents to other document tag profiles to find similar documents.

(4) We can form “document profiles” for a document using the spreading activation method over the bigraph. We compare these document profiles for documents to find similar documents.

Steps
Having described the conceptual ideas behind the algorithm, we now turn to the specific steps of the algorithm. TagSearch is done using a multi-step process:

(Step 1) First, we construct a bigraph between URLs and tagging keywords. Bookmarks in these systems are typically of the form [url, tag1, tag2, tag3, tag4, ...]. We can decompose/transform them into the form [url, tag1], [url, tag2], and so on.

Given tuples in the form [url, tag], we can form a bigraph of URLs linked to tags. This bigraph can be expressed as a matrix. This process is depicted in Figure 3.

(Step 2) Next, we construct “tag profiles” and “document profiles” for each URL and each tag in the system. For each URL and tag in the bigraph, we perform a spreading activation using that node in the bigraph as the entry node.

We can do this, for example, using the bigraph matrix constructed in step 1.

After “n” steps (which can be varied based on experimentation), depending on whether the spreading activation was stopped on the tag side of the bigraph or the document side of the bigraph, we will have a pattern of weights on tags or documents. These patterns of weights form the “tag profiles” or “document profiles”. This process is depicted in Figure 4.

![Figure 3: encoding of the tag/document relationships into a bigraph matrix.](image)

![Figure 4: Spreading Activation of the tag/document bigraph.](image)

Spreading activation have been used in many other systems for modeling concepts that might be related, or to model traffic flow through a website [5]. In this case, we use spreading activation to model tag and concept co-occurrences.

Specifically, the tag profiles and document are computed using spreading activation iteratively as vectors A as follows:

\[
A[1] = E; \\
A[2] = \alpha M \ast A[1] + \beta E; \ldots \\
A[n] = \alpha M \ast A[n-1] + \beta E;
\]
where:

$A[1], A[2], \ldots, A[n]$ are iteratively computed profile vectors of URLs and tags;

$E$ is a unit vector representing a tag or document entry node;

$M$ is a matrix representation of the bigraph arranged by column or row according to the selected entry node;

$\alpha$ and $\beta$ are parameters for adjusting spreading activation.

(Step 3) Having constructed these profiles, we now have several options for retrieval. These profiles form the basis for doing similarity computations and lookups for retrieval, search, and recommendations. For example, for a given document, if we want to find more related document to it, we have three options:

(a) For a document, we can simply lookup the corresponding document profile, and pick the top highest weighted documents in that profile and return that set;

(b) We can also use the corresponding document profile or tag profile and compare it against all other document or tag profiles in the system, and find the most similar profiles and return the matching documents;

(c) If the document is not already in the bigraph, we can first use standard information retrieval techniques (for example, cosine similarity of the document word vectors) to find the most similar document that is in our bigraph, and use method (a) or (b) above to find related documents in our bigraph.

If instead, for a given document, we want to look for related tags to it, we can:

(a) Lookup the corresponding tag profile for that document, and choose the top highest weighted tags in that profile and return that set;

(b) Use the corresponding document/tag profile for that tag and compare it against all other document/tag profiles for other tags in the system, and find the most similar profiles and return the matching tags.

(c) If the document is not already in the bigraph, we can first use a standard information retrieval technique to find the most similar document that is in our bigraph, and use method (a) or (b) above to find related documents in our bigraph.

For a given tag, if we want to find related documents or related tags to it, we can again use similar methods (a) or (b) as described above, if the tag already exists in our bigraph. If the given tagging keyword is not in the bigraph, we can first perform a standard keyword search to find the first initial related documents and tags. We can then further refine the result set by the above methods.

**Multi-word queries**

For any query with multiple tags as the query, the system looks up all of the corresponding document profiles for those tags and simply adds these profiles together to find the most relevant documents. Given that the tag profiles are expressed as spreading activation vectors, addition simply adds the activation values together, creating the equivalent of an OR query on the tag keywords.

For a more strict search simulating an AND query, instead of adding the activation vectors together, we only include a result in the set when the URL has actually been tagged with all of the query keywords.

**Negative Keywords**

For relevance feedback, the user can tell the MrTaggy system that a tag represents concepts that she is not interested in. To handle these ‘negative keywords’, the system, instead of adding the corresponding document profile, it subtracts the corresponding document profile from the activation vector.

**IMPLEMENTATION BASED ON MAP-REDUCE**

Having described the algorithms, we now turn to the

![Figure 5. Overall dataflow and architectural diagram of the MrTaggy implementation of the TagSearch algorithm.](image-url)
description of how we actually implemented the algorithm in a real system. Figure 5 shows an architecture diagram of the overall system we released on the Web called MrTaggy.com.

First, a crawling module goes out to the Web and crawls social tagging sites, looking for tuples of the form <User, URL, Tag, Time>. We keep track of these tuples in a MySQL database. In our current system, we have roughly 140 million tuples.

A MapReduce system based on Bayesian inference and spreading activation then computes the probability of each URL or tag being relevant given a particular combination of other tags and URLs. As described above, we first construct a bigraph between URL and tags based on the tuples and then precompute spreading activation patterns across the graph. To do this backend computation in massively parallel way, we used the MapReduce framework provided by Hadoop (hadoop.apache.org). The results are stored in a Lucene index (lucene.apache.org) so that we can make the retrieval of spreading activation patterns as fast as possible.

Finally, a Web server serves up the search results along with an interactive frontend. The frontend responds to user interaction with relevance feedback arrows by communicating with the Web server using AJAX techniques and animating the interface to an updated state.

In terms of data flow, when the user first issues a query, the Web server looks up the related tag recommendations as well as the URL recommendations in the Lucene index and returns the results back to the frontend client.

The client presents the result to the users with the arrows buttons as relevance feedback mechanisms. When the user presses on one of the arrow buttons, the client issues an updated query to the Web server, and a new result set is returned to the client.

**MRTAGGY BROWSING/SEARCH INTERFACE**

Having just described how the algorithm operates in the backend, we now describe the interaction of the relevance feedback part of the system. Figure 6 shows a typical view of the tag search browser, which is available publicly at MrTaggy.com.

![MrTaggy UI](image)

Figure 6. MrTaggy UI with “search tags” section for added tags and “bad tags” section for excluded tags (both on the left).
The left side of the figure shows how the system displays the recommended tags. The right side lists the top document recommendations based on the input so far.

MrTaggy provides explicit search capabilities (search box and search results list) combined with relevance feedback [1, 21] for query refinements. Users have the opportunity to give relevance feedback to the system in two different ways:

Related Page Feedback: By clicking on the downward arrow a search result can be excluded from the results list, whereas by clicking on the upward arrow the search result can be emphasized which leads to an emphasis of other similar Web pages.

Related Tag Feedback: The left of the user interface presents a related tags list (see Figure 6), which is an overview of other tags related to the relevant keywords typed into the search box. For each related tag, up and down arrows are displayed to enable the user to give relevance feedbacks. The arrows here can be used for query refinements either by adding a relevant tag or by excluding an irrelevant one.

In addition, users can refine the search results using tags associated with each of the search results. During search, result snippets (see Figure 7) are displayed in the search results list. In addition to the title and the URL of the corresponding Web page, instead of a short summary description, a series of tags are displayed. Other users have used these tags to label the corresponding Web page. When hovering over tags presented in the snippet, up and down arrows are displayed to enable relevance feedbacks on these tags as well.

Users’ relevance feedback actions lead to an immediate reordering or filtering of the results list, since the relevance feedback and the search result list are tightly coupled in the interface. We use animations to display the reordering of the search results, which emphasizes the changes that occurred in the result list (see Video at http://www.youtube.com/watch?v=gwYbonHi5ss). New search results due to the refinements are marked with a yellow stripe.

Quick Links: A recently added feature of the interface is the mashup of quick search results from the Yahoo! BOSS search API. In parallel with the TagSearch process, we issue a Yahoo! Search to obtain the top 1-2 results and display those results as quick links. In this way, if our bookmarks do not offer any good suggestions, the user could use these quick links to get started on their topic explorations as well.

SUMMARY OF THE EVALUATION
We recently completed a 30-subject study of MrTaggy and Kammerer et al. describes the study in detail [15]. In this study, we analyzed the interaction and UI design. The main aim was to understand whether and how MrTaggy is beneficial for domain learning.

We compared the full exploratory MrTaggy interface to a baseline version of MrTaggy that only supported traditional query-based search. In a learning experiment, we tested participants’ performance in three different topic domains and three different task types. The results are summarized below:

(1) Subjects using the MrTaggy full exploratory interface took advantage of the additional features provided by relevance feedback, without giving up their usual manual query typing behavior. They also spent more time on tasks and appear to be more engaged in exploration than the participants using the baseline system.

(2) For learning outcomes, subjects using the full exploratory system generally wrote summaries of higher quality compared to baseline system users.

(3) To also gauge learning outcomes, we asked subjects to generate keywords and input as many keywords as possible that were relevant to the topic domain in a certain time limit. Subjects using the exploratory system were able to generate more reasonable keywords than the baseline system users for topic domains of medium and high ambiguity, but not for the low-ambiguity domain.

Our findings regarding the use of our exploratory tag search system are promising. The empirical results show that subjects can effectively use data generated by social tagging as “navigational advice”. The tag-search browser has been shown to support users in their exploratory search process. Users’ learning and investigation activities are fostered by both relevance feedback mechanisms as well as related tag suggestions that give scaffolding support to domain understanding. The experimental results suggest that users’ explorations in unfamiliar topic areas are supported by the domain keyword recommendations and the opportunity for relevance feedback.

CONCLUSION
For exploratory tasks, information seeking systems should focus on providing cues that might make these explorations

5 Ways to Mix, Rip, and Mash Your Data

mashup web2.0 pipes mashups tools programming webapps as webservices aggregator

http://www.techcrunch.com/2007/03/02/5-ways-to-mix-rip-and-mash-your-data/

Figure 7. The 3 parts of a search result snippet in the MrTaggy interface: title, tags, URL.
more efficient. One possible solution is building social information seeking systems, in which social search systems utilizes social cues provided by a large number of other people. Social bookmarks provide one such exploratory cue that systems can utilize for navigational signposts in a topic space.

In this paper, we described the detailed implementation of the TagSearch algorithm. We also summarized a past study on the effectiveness of the exploratory tool. Since social search engines that depend on social cues rely on data quality and increasing coverage of the explorable web space, we expect that the constantly increasing popularity of social bookmarking services will improve social search browsers like MrTaggy. The results of this project point to the promise of social search to fulfill a need in providing navigational signposts to the best contents.

REFERENCES


ABSTRACT
Collaborative filtering (CF) is a method for personalized recommendation. The sparsity of rating data seriously impairs the quality of CF’s recommendation. Meanwhile, there is more and more tag information generated by online users that implies their preferences. Exploiting these tag data is a promising means to alleviate the sparsity problem. Although the intention is straightforward, there’s no existed solution that makes full use of tags to improve the recommendation quality of traditional rating-based collaborative filtering approaches. In this paper, we propose a novel approach to fuse a tag-based neighborhood method into the traditional rating-based CF. Tag-based neighborhood method is employed to find similar users and items. These neighborhood information helps the sequent CF procedure produce higher quality recommendations. The experiments show that our approach outperforms the state-of-the-art ones.

ACM Classification Keywords
H.3.3 Information Storage and Retrieval: Information Search and Retrieval-Information filtering

General Terms
Algorithms, Experimentation

Author Keywords
Tags, Latent Dirichlet Allocation (LDA), Collaborative Filtering, Neighborhood Method

INTRODUCTION

Nowadays people are inundated by choices. Personalized recommendation is a solution to this problem. Various kinds of recommender systems are employed for better user experience. Collaborative filtering [4, 12] is one of the best techniques of choice therein. This technique tries to identify users that have relevant interests by calculating similarities among user profiles. The idea is that it may be of benefit to one’s search for information to consult the behavior of other users who share the same or relevant interests.

Because collaborative filtering recommendation depends on the preference of the users with the same or relevant interests, the similarity computation imposes significant influence on the quality of recommendation. Early item-based and user-based collaborative filtering approaches find similar users or items (neighbors) by comparing the rating records of different users or items cannot help to find the best neighbors. If a user has few ratings for items or this user only gives all his/her ratings to the unpopular ones, it will be difficult for those approaches to find the proper neighbors.
tract sufficient feature information, which reflects the problem of data sparsity. It is because they fit the original matrix by feature extraction only based on the rating data while the rating data are extremely sparse. If we could obtain more ratings, we would surely enhance the quality of fitting process. From this standing point, we propose a better collaborative filtering approach to exploit additional knowledge from the tags as a supplement to ratings.

Tags are simple, ad-hoc labels assigned by users to describe or annotate any kind of resource for future retrieval. Their flexibility means they probably capture a user’s perspective and preference with ease. Most recent work focuses on the tag recommendation in which the objects to recommend are tags [18, 20, 22, 27]. In the case of item-based recommendation, users expect to get specific suggestion on which item might be interesting. There are a limited number of solutions for this situation, and most of them do not have a generalized adaptation to different data resources because they ignore abundant rating data [11, 25]. In this paper, we offer a novel personalized recommendation method which matches the case of containing both ratings and tags.

Our approach still shares the main idea of classic neighborhood method, but there are some differences in where to find neighbors. The neighbors are usually found in the ratings for the traditional CF approach [1]. We do not find neighbors directly by this means. First we exploit the latent topic grouping information hidden in tags and then we find groups of the users interested in similar topics and collections of the items under similar topics. To predict the user’s rating for the item, we consult the ratings of both of the user’s and the item’s neighbors by employing a neighborhood method. Thanks to taking into account both tag neighbors and rating neighbors, our method outperforms most popular CF approaches.

The structure of the rest of the paper is as follows. In Section 2 we introduce the background and the related works. In section 3 we explain our improved collaborative filtering method in details. In Section 4 we give two toy examples and compare our method with NMF, PMF and SVD on a popular movie dataset. And finally we conclude this paper with some future work.

PRELIMINARIES
Rating prediction is one of the most popular means to evaluate the performance of collaborative filtering algorithms. From the rating data of most collaborative filtering datasets, we can obtain a N × M rating matrix R including N users and M items. Matrix R is defined as

\[ r_{ij} = \begin{cases} \text{user}_i\text{'s rating for item}_j, & \text{if user}_i \text{ has rated item}_j \\ 0, & \text{otherwise} \end{cases} \]

where \( i \in \mathbb{N}^+ \), \( j \in \mathbb{M}^+ \) and \( r_{ij} \in [1, R_{max}] \). The usual evaluation process is the hold-out cross validation [5]. A certain proportion of ratings are hidden for testing and the rest are used for training. The measures of evaluation include complexity and accuracy. Nevertheless, the accuracy is much more important because most of the Collaborative Filtering approaches are offline. Therefore, it is the focus in this paper.

Naive Estimates
One of the most instinctive predicting methods is to compute the mean values. Taking the user’s and the item’s average biases involved, we get the naive estimate [8]:

\[ b_{ij} = \mu + b_i + b_j, \]

where \( b_{ij} \) indicate the predicted rating of \( \text{user}_i \) on \( \text{item}_j \); \( \mu \) is the global average rating; \( b_i \) and \( b_j \) denote \( \text{user}_i \text{'s and item}_j \text{'s average bias, respectively.} \)

This naive method is effective and scalable, but it does not take the interaction between users into account. Every user’s rating for a item has influences on other users’ opinions to that item. This interdependence between the users forms a social network [24] which connects all users together. The personalized recommendations are not delivered in isolation, but in the context of this social network [14]. The neighborhood method is one of the most effective methods to analyze the context.

Neighborhood Method
The aim of the neighborhood method [2] is to find the users who give similar ratings and the items which receive similar ratings. The approximate ratings infer the potential similarity of the future ratings. This is the basic assumption of collaborative filtering. Because the neighborhood method digs out from the neighbors the clues that indicate the potential ratings, it produces better predictions than the naive estimate. The model of the neighborhood method unifying item-based and user-based collaborative filtering approaches is

\[ \hat{r}_{ij} = b_{ij} + \sum_{h \in S^k(i;j)} \theta^i_{h}(r_{ih} - \bar{b}_h) + \sum_{h \in S^k(j;i)} \theta^j_{h}(r_{hj} - \bar{b}_h), \]

where \( \hat{r}_{ij} \) is the predicted rating; \( b_{ij} \) refers to the naive estimate’s prediction; \( S^k(j;i) \) denotes the set including k nearest rated items neighboring with \( \text{item}_j \) for a given \( \text{user}_i \) and \( r_{hj} \); \( S^k(i;j) \) denotes the set including k nearest users neighboring with \( \text{user}_i \) for a given \( \text{item}_j \) and \( r_{ih} \). \( \theta \) reflects the different weights of \( r_{ih} \).

There are several representations for the weights. The cosine similarity is one the most effective measures to indicate the different weights.

\[ \text{Cosine Similarty} = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2}, \]

where \( a \) and \( b \) are both vectors with the same dimension.
The neighborhood method finds item $j$’s $k$ nearest neighbors (k-NN). These neighbors infer the potential value of $r_{ij}$ to different degrees according to their similarity with item $j$. Although there are several different similarity measures employed to compute the similarity between the items, the similarity between items is represented by the distance between their rating vectors. The similarity of the items which have less common raters are structurally lower. If there’re high level features extracted to represent the user and the item, the similarity can be better measured this way. Matrix factorization methods learn this lesson.

**Matrix Factorization**

To extract high level features, matrix factorization methods try to find the rating matrix’s low rank approximations [15, 21]. They focus on fitting the user-item rating matrix by low-rank approximation and use the fitting result to make subsequent predictions [6, 7, 8, 16]. The premise behind this low-dimensional factor model is that there is only a small number of factors or features influencing preferences, and that a user’s preference for an item is only determined by that user’s feature vector and that item’s feature vector.

What is related to our work is not the basic matrix factorization methods. Recently, some matrix factorization methods which involve auxiliary information analysis draw our attention. [13] proposes a trust-aware collaborative filtering algorithm. The algorithm is based on the general knowledge that people normally ask friends for recommendations. Due to the memory-based model, this algorithm suffers from huge online costs. Trust values need to be computed like similarity measures. SoRec [10] fuses the existed trust-based approach with Probabilistic Matrix Factorization (PMF) [16]. This method is model-based, but it cannot be widely applied due to the scarce resource of trust information which involves people’s privacy. [9] proposes a relation regularized matrix factorization method for relational data analysis. Yet it is designed for making recommendations concerning objects that have both content and links. The idea of Collective Matrix Factorization [19] is innovative: factorizing multiple matrices simultaneously with shared parameters. The weakness of this method is that the parameter learning process is computationally costly.

**TAG-BASED ITEM RECOMMENDATION**

Since tags and ratings are two of the most attributes attached to items, we propose a generalized neighborhood recommendation method to make use of them in the same time. Our work is based on the assumption that the behavior of tagging and rating share the same motivation: item classification. In this sense, the latent preference information found in tagging data has more power than that in rating data. Regarding tags, there are two types of recommendation: item recommendation and keyword recommendation. Our concern is item recommendation which is the same with most CF recommendation methods. In the background of electronic commerce and video on demand, proper item recommendations are better since the items are overwhelmingly numerous.

**Topic Finding**

As with the rating data, the tag data can be represented as a $n \times m$ sparse matrix $T$ given $n$ users and $m$ items, $\{\text{user}_i$’s tags for item $j$, if $\text{user}_i$ has tagged item $j \}$, null, otherwise.

The users are allowed to give more than one tag to each item. So if the tags are clearly separated, $T$ becomes a three-dimensional tensor. The three dimensions are user, item, and tag. This is a tough case to take care of and it is why there is little work on extract preference information from this data resource. Innovatively, we divide $T$ into user-tag and item-tag matrices representing the tags given by the users and the tags received by the items, respectively. The user-tag and item-tag matrices are denoted as $T^U$ and $T^I$ which are defined as follows:

$$T^U = [t_{1,1}, t_{2,1}, ..., t_{n,1}]^T,$$

$$T^I = [t_{1,2}, t_{2,2}, ..., t_{n,2}]^T,$$

where $t_u$ denotes the tags $\text{user}_u$ has given, and $t_i$ denotes the tags $\text{item}_i$ has been given.

When the original tensor $T$ is converted into bags of words in $T^U$ and $T^I$, we can apply LDA [3] to find latent topic information therein. Processing $T^U$ and $T^I$ separately, we find their latent topics in the form of probability.

$$\theta_i^j = p(\text{topic} = j | \text{user} = i),$$

$$\theta_j^i = p(\text{topic} = j | \text{item} = i),$$

where $\theta_i^j$ denotes $\text{user}_i$’s probability of preferring for topic $j$, and $\theta_j^i$ denotes $\text{item}_i$’s probability of being related to the topic $j$. This is a kind of “soft clustering”. It is possible that a user or item is under multiple topics with the same probability. The similarity between these row vectors in $\Theta^U$ and $\Theta^I$ more appropriately reflects the users’ and items’ similarity because clustering is based on the semantics of the tags. The matrices $\Theta^U$ and $\Theta^I$ are not directly used for rating prediction. Because they are full matrices which are not appropriate for computation, we set a threshold value to reserve high similarity relations and clear the others. Another important reason for this process is that most of the users and items are indirectly related with each other in reality.

After finding the matrices $\Theta^U$ and $\Theta^I$, it is easy to employ k-NN clustering to find the groups whose members share the same interests or attributes.

**Rating Prediction**
We assume all the users who tag the items also give ratings and that all the items which are tagged also receive ratings. If some users actually fail to give either ratings or tags, we still can make use of what they input to the recommender system. Even with few entries, the recommender system still understands what the user wants. We hold this claim because tags are more informative. If the user only put one tag “anime” to some item, we could infer that this is an animation fun. But if this user only give a high rating to the movie "Avatar", what should we infer from this? Which groups of movie does this user like, actions, adventures, fantasies or sci-fis?

Most recommender systems use the integral interval $[1, R_{max}]$ to represent the users’ preference on items. It is necessary to normalize the ratings into the range to $[0, 1]$, because only this interval makes sense for probability. There are many mapping methods. One of the most widely used mapping function is $f(x) = (x - 1)/(R_{max} - 1)$. As far as we know, the influence exerted on the final recommendation by different mapping functions is not significantly different.

So our next step is to make rating predictions based on the grouping results stated in the last section. The prediction is made according to a neighborhood method:

\[
\hat{r}_{ij} = \mu + b^*_i + b^*_j,
\]

\[
b^*_i = \frac{\sum_{h \in T(i)} (r_{ih} - b_h)i_R^h}{\sum_{h \in T(i)} i_R^h},
\]

\[
b^*_j = \frac{\sum_{h \in T(j)} (r_{jh} - b_h)j_R^h}{\sum_{h \in T(j)} j_R^h},
\]

where $b^*_i$ denotes user i’s bias for the topic $T(i)$, and $b^*_j$ denotes item j’s bias for the topic $T(j)$. $T(i)$ and $T(j)$ denote the topic user i is interested in and the topic item j is under, respectively. Each topic is a set which contains a number of users or items.

Plus, we give different weights to the neighbors with different distances. The algorithm’s weighted variant is

\[
\hat{r}_{ij} = \mu + b^*_i + b^*_j,
\]

\[
b^*_i = \frac{\sum_{h \in T(i)} (r_{ih} - b_h)i_S^h i_R^h}{\sum_{h \in T(i)} i_S^h i_R^h},
\]

\[
b^*_j = \frac{\sum_{h \in T(j)} (r_{jh} - b_h)j_S^h j_R^h}{\sum_{h \in T(j)} j_S^h j_R^h},
\]

where $\theta_i$, $\theta_i$, and $\theta_j$ denote the row vectors of probabilities in $\Theta$. $S$ represents the cosine similarity of the vectors $\theta_i$ and $\theta_j$.

**EXPERIMENTAL ANALYSIS**

**Dataset Description**

Movielens Dataset is created by Movielens movie recommender which aims to provide online movie recommendation service [17]. Their work is a more involved system rather than a particular algorithm, so we do not delve into it. Their dataset includes three files. One is rating data which contains users’ ratings for movies, another is tagging data which contains movies’ tags and the user’s id who made the tag, and the other is movie overview data which contains the movie’s name, release year and genre. The user are allowed to give ratings and tags to the movies they have seen. The ratings are integers between 1 to 5.

We intend to leverage tag analysis to help rating prediction. So we need the movies that have both ratings and tags and the users that give both ratings and tags. After taking the intersection of the rating data and tag data, we get the rating data’s subset which contains all the tagged movies. This subset contains 905686 ratings from 4001 users for 7600 movies. The density of these rating data is

\[
\frac{905686}{4001 \times 7600} = 2.974%.
\]

From this subset, we randomly and independently choose 20%, 50%, 80% and 99% of rating data as separate training sets. The remaining rating data are used as the testing sets. The experiments are all repeated five times to reduce errors.

**Toy Examples**

Because the quality of recommendation is eventually reflected in the results of rating prediction accuracy. To obtain a clearer vision about the qualitative quality, we present two toy examples in smaller data volume scale.

First we extract the tags from 6 users. The tag matrix $T^U$ is as follows:

\[
\begin{pmatrix}
\text{horror} & \text{killer} & \text{action} & \text{horror} \\
\text{horror} & \text{action} & \text{thrill} & \text{action} \\
\text{anime} & \text{Japanese} & \text{fantasy} & \text{anime} \\
\text{documentary} & \text{911} & \text{terrorist} & \text{hero} \\
\text{historic} & \text{documentary} & \text{American} & \text{realistic}
\end{pmatrix}
\]

We set the hyper-parameter topic number as 3 and conduct LDA analysis to get the probabilistic matrix

\[
\Theta^U = \begin{pmatrix}
0.345679 & 0.345679 & 0.308642 \\
0.364198 & 0.308642 & 0.308642 \\
0.308642 & 0.345679 & 0.345679 \\
0.327160 & 0.345679 & 0.327160 \\
0.345679 & 0.308642 & 0.345679 \\
0.308642 & 0.345679 & 0.345679
\end{pmatrix}
\]

It is quite obvious that user$_1$ and user$_2$ have the same or similar interests. The first column values is the maximum among all three columns for both of him, which infers the topic they are most probably interested in is the first topic. For the same reason, user$_3$ and user$_4$
are similar and users\textsubscript{5} and users\textsubscript{6} are alike. The subjective grouping result is exactly the same with that of a k-NN method to the matrix $\text{Theata}^U$ with the same hyper-parameters. Considering the semantic meanings of these 24 tags, we conclude that the result of the analysis on the user-tag matrix is persuasive. The analysis on the item-tag matrix is effective in a similar way. Yet, there are more merits for analyzing the item-tag matrix because we can obtain the names and genres of the movies from the overview data of the dataset.

In the second example, we extract the tag data in MovieLens dataset and get the movie-tag matrix. There are 7600 movies in this matrix. We make the tag analysis to find the latent topics again. For the sake of leaving more space for showing more important results, we set the desired topic number as 10. Fig.1 presents the five most probable movies per topic. According to it, we have several observations as follows:

1. There are at least 4 movies under the same genre for every topic. Topic\textsubscript{1}, Topic\textsubscript{2} and Topic\textsubscript{3} all have more than two such common genres. The high co-occurrence frequency of different movies under the same genre reflects the large extent of conformation. The coexistence of movie series like *The Matrix* and *Star Wars* under Topic\textsubscript{7} illustrates our tag analysis can find not only the movies of the same genres but also the movies of the same series.

2. The five movies in Topic\textsubscript{6} all reflect big social problems. This problem could be war, terrorist attack, or social security crisis. This explains why these movies under different genres are in the same topic. The topic finding result is not equal to the genre classification. The topic "social problem" may be interesting for some of the users. These details are more valuable for inferring the users' preference than genres.

3. According to the corresponding rating data, the average variance of ratings of the five movies under their
corresponding topic is just 0.102. It illustrates that users hold similar preferences for the movies with similar probability under each topic. So we posit that consulting the movies with similar probability under each topic can help improve personalized rating prediction.

**Metrics**

We use the Root Mean Square Error (RMSE) metrics to measure the prediction quality of our proposed approach in comparison with other collaborative methods. RMSE is defined as:

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{N} \sum_{j=1}^{M} (r_{ij} - \hat{r}_{ij})^2}{N M}}
\]

where \( r_{ij} \) and \( \hat{r}_{ij} \) are the actual rating and predicted rating from \( N \) users for \( M \) movies. Plus, we use rounded value as our predicted rating. The errors of prediction with rounded rating value are more obvious. But whether the rating is rounded or unrounded, the comparison results between different approaches does not change a lot.

**Comparison**

We compare our approach with two collaborative filtering algorithms: Non-negative Matrix Factorization (NMF) method, PMF method and the improved regularized SVD method. In fact, one of the most difficult problems in our work is to find some coordinate algorithms for comparison. Because our intention is to provide a generalized item recommendation model to combine the use of ratings and tags, most of the related work is inapplicable to the data resource in this situation. We choose three of the most popular algorithms by expediency.

The parameters of these two methods also need to be tuned. According to the relative works and our experiments, the best parameters for the PMF approach on Movielens dataset are like these: \( \lambda_u = 0.001, \lambda_v = 0.0001 \). Concerning the improved regularized SVD method, \( lrate = 0.001, \lambda = 0.02, \lambda_2 = 0.05 \). We set the number of feature dimensions as 80. We think this assignment is reasonable because the commonly used feature dimension for these matrix factorization is between 30 and 100.

We have six versions of the improved collaborative filtering methods. Nghbru represents the neighborhood recommendation method based only on the user-tag analysis. Nghbri corresponds to the variant based only on the item-tag analysis. Nghbra integrates the use of the user-tag analysis and the item-tag analysis. Each of these three methods, there are two different weighting strategies. One is to use uniform weights, labeled as “Avg”; the other is to use different weights, labeled as “Wgt”.

<table>
<thead>
<tr>
<th>Percentage</th>
<th>RMSE Nghbru</th>
<th>RMSE Wgt Nghbru</th>
<th>RMSE Nghbri</th>
<th>RMSE Wgt Nghbri</th>
</tr>
</thead>
<tbody>
<tr>
<td>20%</td>
<td>0.8811</td>
<td>0.8807</td>
<td>0.8803</td>
<td>0.8803</td>
</tr>
<tr>
<td>50%</td>
<td>0.8799</td>
<td>0.8796</td>
<td>0.8789</td>
<td>0.8786</td>
</tr>
<tr>
<td>80%</td>
<td>0.8792</td>
<td>0.8791</td>
<td>0.8791</td>
<td>0.8786</td>
</tr>
<tr>
<td>99%</td>
<td>0.8790</td>
<td>0.8790</td>
<td>0.8788</td>
<td>0.8786</td>
</tr>
</tbody>
</table>

The results in Table 1 show our neighborhood recommendation method outperforms the improved regularized SVD method more than 41%, NMF 36%, and PMF 23%. We would like to analyze the results more specifically: 1) For all these algorithms in Table 1, the prediction accuracy increases as the training set’s percentage ascends. This is reasonable because with high training data percentage, our algorithms find more neighbors to consult. The more neighbors we find, the more accuracy we get. 2) Among our own several methods, the version Nghbra-Wgt presents the best performance. It illustrates that utilizing all the tag information and assigning different weights to this tag information is meaningful. 3) We also observe that item tag analysis is a little more effective than the user tag analysis. Although the difference is subtle, it explains the fact that the item-based collaborative filtering approaches are more popular than the user-based ones in early works. 4) Besides, we find the performance increase of Nghbru is obvious compared with Nghbru and Nghbri. This illustrates that the fusion of the user tag analysis and the item tag analysis is lossless. 5) Nevertheless, the performance of the weighted version of Nghbru, Nghbri and Nghbra is not much better than their average counterparts. This can be explained by the homogeneity of users. There are no authorities to give the overwhelmingly important rating. The phenomenon reflects the democracy in the online social network.

**Parameter Analysis**

For topic finding, we set the Dirichlet priors \( \alpha \) and \( \beta \) to 50/\( K \) and 0.1, respectively (\( K \) is the number of topics). These two hyper-parameters are the empirical values for LDA. The threshold value of processing probabilistic matrices \( \Theta_U \) and \( \Theta_I \) is set as 0.03 which means statistically impossible. The other two parameters, iteration number and topic number are unfixed. We explore the optimal solutions for them. Because the parameters of topic finding are different regarding the objects to analyze, we separate the process of tag analysis into user-tag analysis and item-tag analysis. We observe a shape RMSE increases with huge vibrations after 340 iterations for user-tag analysis. This can be seen as the signal of overfitting. Regarding the item-tag analysis,
we observe the optimal iteration number is 480. So we think the optimal iteration number for them is 340 and 480, respectively. And we use these two parameters in the above experiments.

Figure 2. Dependence between the topic number and the prediction accuracy for the items’ tag analysis

Figure 3. Dependence between the topic number and the prediction accuracy for the users’ tag analysis

Compared with the iteration number, we are more interested in the dependence between the topic number and the prediction accuracy. We record the effects to the prediction preciseness for the topic numbers ranging between (1, 50). From Figure 2 we observe that the optimal topic number for the items’ tag analysis is less than 25. The optimal value does not necessarily mean the global optimal value. Although just local optimum, the RMSE value near the topic number of 25 is stably less than 0.9. The observation that there are some better parameter choices less than 10 can be explained by the data sparsity. Rating data sparsity is always existent and thus there are possibilities that our algorithm cannot find enough consultants for potential rating inference when the topic number is set relatively large. One probable case is like this: the neighbors found by our approach is very similar to the movie we are to predict, but the user has not given a rating to it.

For user-tag analysis, the optimal topic number is 23 as Figure 3 illustrates. The situation here is similar with that in item-tag analysis. The local minimum is not the global one. The reason for this is the same as mentioned before. What is different is the stable duration here is shorter than that in item-tag analysis and the fluctuation here is more obvious. This observation can be explained by the diversity of personal interests. Compared with movies, the attributes of human beings are more dynamic and diverse. It is easier to find similar items than similar people because the measure in the latter situation is vaguer.

In summary, the optimal topic number is around 25 in both two situations, which means the results are consistent. And the genre number in common use is the same order of magnitude. From this perspective, our results are reasonable. But we must emphasize the fact that our method of topic finding and the common use of genre classification focus on different targets and thus produce different results. Considering the fact that the information of genre demands the knowledge from experts, our method of topic finding has wider range of application.

Discussion

There are some issues concerning implemental details we need to explain her. Because tags are given with high freedom, there are a lot of preprocessing work to do. First, there are many noise such as “:D” in the data. We absolutely should remove them all. But in fact, we are unable to guarantee all noise are cleared off because they lack rules. Second, stoplist is one of the most important parts to manipulate. To our best knowledge, most stoplists used in document clustering remove the word “good” and “great”. Concerning our approach, these words reflect the users’ preference to the items. It is somehow meaningful. We hesitate to remove these words because it may benefit rating prediction. In our experiments, we remove the prepositions, conjunctions and other less meaningful words while leave emotional words untouched. Third, stemming is also a complicated technique that we must employ. It may be simpler for ordinary document and webpage retrieval. But the bags of tags in our experiment are rather chaotic. We are worried the stemming algorithm may to some extent have a negative influence on the quality of topic finding. If we take better care of these three factors, we should further improve the quality of recommendation to some extent.

CONCLUSIONS AND FUTURE WORK
In this paper we proposed a novel method to alleviate the sparsity problem and improve the quality of the collaborative filtering method. We make use of the tag information to find closer neighbors for the users and the items, respectively. These neighbors give strong inferences for the potential preference of the user for the item. We utilize these inferences to make the rating prediction. According to the experiments, our approach's prediction for the users' preference is much more accurate than the art-of-state ones such as NMF method, PMF method and the improved regularized SVD method.

Finding neighbors is vital for an excellent collaborative filtering algorithm. The motivation of our work is to find better neighbors, which give stronger inference for the prediction. Latent topics connect the users and items with similar interests together. Finding these topics is equal to finding the neighbors. The connection that cannot be discovered in the rating records can be disclosed through learning about the tagging history. This is why our method outperforms the others.

The next step for us is to improve our method in two aspects. One is incrementalization. The neighborhood method enjoys the low computational complexity, but suffers from the rigidity to frequent update. If there are a lot of new entries from the users to items, our current solution fail to deal with this situation. On the other hand, we can fuse the collaborative matrix factorization method with our topic finding model. This is another way to make use of the latent topic information. We believe these methods are both promising solutions to further improve the collaborative filtering technique.

ADDITIONAL AUTHORS

REFERENCES


Ethical intelligence
in social recommender systems

Peter Juel Henrichsen
Center for Computational Modelling of Language
Copenhagen Business School
pjh.isv@cbs.dk
+45 3815 3130

ABSTRACT
We present a new algorithm suitable for social recommender systems connecting web-clients based on their personal characteristics and interests. The algorithm has been tested in a web-based application with a conversational agent serving as a sophisticated switchboard operator. Based on a short client-agent conversation, the profile of the client is determined and compared to the profiles stored in a database. The current system is focused on the overall topic "ethical choices in life", demonstrating simple formal principles and computational tractability even within a notoriously difficult semantic domain. The algorithm is radically data-driven, informed by a background corpus of automatically collected text documents. The system is robust towards irregular input (misspellings, broken syntax), and this fact in combination with fine computational scaling properties could make it attractive for implementation in mass-services like Facebook and the blog-world. Freddy will be available for test runs at SRS-2010.

Author Keywords
Social Recommender Systems; Client matching; Man-machine interface; Ethics; Conversational agent

ACM Classification Keywords
Intelligent agents; Query processing; Relevance feedback; Information Search and Retrieval; Selection process

INTRODUCTION
Joseph Weizenbaum's classical program ELIZA [1] left the impression on many people that a machine could actually understand the psychological reality of a human.

Eliza:- Please state your problem.
Client:- I'm sad
Eliza:- I am sorry to hear that you are sad.
Client:- my wife just left me
Eliza:- Tell me more about your family.

Inspired by ELIZA, we developed a web-based conversational agent with a flair for ethics. The agent, Freddy, was intended as a sophisticated switchboard operator connecting web-clients of similar ethical inclination. The discourse planning system uses a simple psychological fact: you can learn a lot about your conversational partner by just exchanging a few remarks.

Freddy:- What do you do to improve the world?
Client:- I'm always riding my bike to work
Freddy:- Instead of taking your car?
Client:- yeah - better for the environment
Freddy:- What else do you do for the environment?
Client:- I use green energy, like wind generated electricity; I also use energy saving light bulbs
Freddy:- Do you wish to chat with another person who is also conscious of her carbon emission footprint?

The coherence in this conversation is based on a common ground concerning the ethical relevance and impact of the actions mentioned. In this case, Freddy discovered that "wind generated electricity" is associated with "carbon emission footprint" within the realm of sustainability. This kind of association and reasoning is not difficult for humans. Consider an example.

(1) I always use a push lawn mower
(2) Riding my bike to work
(3) Avoid dairy as much as possible
(4) Animal flesh is not for eating
(5) Only use green energy sources
(6) Do not kill and consume living creatures

People asked to label statements 1-6 with either environmentalism or vegetarianism according to their ethical motivation, uniformly pick (1), (2), and (5) as e, (3), (4), (6) as v. In spite of the huge linguistic and practical diversity of (1)-(6), people find the classification task easy.

Ethical terms that are easy to apply, may however be hard to define. Lexical definitions of ethical concepts tend to be expressed in other terms that are just as vague and abstract. By way of example, "solidarity" is explained by Wikipedia as "a unity of purpose or togetherness", and "social solidarity" as "the integration, and degree and type of integration, shown by a society or group with people and their neighbors". Such definitions seem next to impossible to fit in a rule-based reasoning system. Given our focus on ethically based match-making, we therefore opted for a
data-driven (as opposed to rule-driven) approach. In our system, decision-making is thus based solely on statistical analyses of human actions and discourse as represented in a large background corpus.

In this paper we present those components of the system that are contributing to Freddy's ethical flair. A demo will available for test-runs at SRS-2010.

**FREDDY'S VITAL PARTS**

For readability, we use *ethical quality terms, EQTs* and *subsections of the Corpus* interchangeably. By a type, we refer to a lexical word form while a *token* is a particular occurrence of a type. We speak of *large* (*small*) *tokens*, meaning tokens with relatively many (*few*) occurrences.

The Corpus

The Corpus represents the system's world knowledge. It is subdivided in sections each representing a central theme. The global set of themes must (i) contain only very general terms, (ii) define the overall topics of the conversational realm. The determination of the theme set is the only non-automatic part of the preparation and maintenance of the client matching system. Currently, Freddy's conversational realm is defined by 23 terms (EQTs).

\[
\text{EQTs} = \{\text{solidarity}', \text'benevolence}', \text'sustainability}', \ldots\}\n\]

The current system uses a Corpus harvested with Google. Each EQT was installed as a search keyword. Visited html pages were stripped for everything but plain text. Long documents were pruned (global max = 5k tokens/doc), and all text lines preceding (following) the first (last) occurrence of the key word were skipped. Remaining tokens were normalized (non-alphabetic characters deleted, etc). Texelines with 2+ occurrences in a single EQT were only represented once in the Corpus (removing multiple copies of the same texts and paragraphs). Otherwise no filtering or token analysis was used.

*Corpus samples*

(from EQT "Solidarity"):

"(...) The power structure of course is still controlled by white men. But the rise of a middle class of all races is real (...)"

"(...) Leave the factory to go to meetings and demonstrations against the war strike against the state sponsored violence. Encourage your enlisted men in the armed forces to do the same (...)"

(from EQT "Veganism"):

"(...) As far as the hot-dog and fanta comment Kelly made that is pointless unless she feels that all meat-eaters approach food this way(...)"

"(...) What about school systems that get the green light for defining corn-syrup filled ketchup as a vegetable? (...)"

Most EQTs in the current Corpus consist of 100k to 300k tokens. The actual sizes and their relative differences have minimal impact on the ethical scorings, as long as each EQT is large enough to avoid sparse-data problems (more on this shortly). A practical lower limit is, say, 50k tokens per EQT.

The Actions database

The Corpus is supplemented by a database of actions, submitted by human clients on various occasions. While the initial Actions Database consisted of 100 actions manually inserted by the author, since then Freddy has been accumulating the actions presented to him by his conversation partners. Currently the Actions Database contains more than 800 actions submitted by approximately 150 agents. Each action is stored with a reference to the originating agent and a canned ethical profile (see next section) for computational efficiency. The Actions Database is used by Freddy for match-making among his human conversational partners based on their ethical profiles.

The ethical profile

Freddy's reasoning is based on a central data-structure called the ethical profile (EthPro), a function mapping a list of input tokens I (typically a description of an action) onto a set of pairs <Q,V>, where Q is an EQT and V the related score value (a positive real number).

\[
\begin{align*}
(A) \quad & \text{EthPro("eat less meat") = } \\
& \text{Vegetarianism} \quad 1.243 \\
& \text{Veganism} \quad 0.954 \\
& \text{Animal rights} \quad 0.863 \\
& \text{Kindness to animals} \quad 0.734 \\
& \text{Environmentalism} \quad 0.583 \\
& \text{Spirituality} \quad 0.454 \\
& \ldots \\
(B) \quad & \text{EthPro("disobey your rulers") = } \\
& \text{Obedience} \quad 0.560 \\
& \text{Anarchism} \quad 0.373 \\
& \text{Strength of character} \quad 0.286 \\
& \text{Privacy} \quad 0.253 \\
& \text{Humbleness} \quad 0.201 \\
& \text{Responsibility} \quad 0.199 \\
& \ldots
\end{align*}
\]

As shown in (A) and (B), EthPros are typically sorted by V.

Observe in (B) that "disobey your rulers" picks Obedience as its most salient EQT. This proposition of course does not exemplify obedience, but the opposite. This illustrates a feature of the Ethical Scoring regime, the *topic* of Obedience being highly relevant in discourse on attitudes towards rulers. Similarly, "meat" and "vegetables" show an affinity to Vegetarianism while food items like "eggs" and "fish" score relatively higher for Veganism (reflecting the distinguishing food items in the particular diets).

Ethical score values are computed as the product of two factors, Global Heterogeneity (GloHet) and Local Homogeneity (LocHom). GloHet has a high value for types unevenly distributed over the Corpus as a whole, for instance when most occurrences are concentrated within a few subsections. GloHet thus measures the capacity of a type as an ethical discriminator.
LocHom measures the distribution of a type within an EQT. A high LocHom value for token \( T \) in section \( Q \) indicates that \( T \) is equally frequent in many or all documents in \( Q \), in other words that \( T \) belongs to the common vocabulary of this particular EQT.\(^2\)

\[
\text{GloHet}_c(T) = \frac{SDev(T,C)}{\text{Mean}(T,C)}
\]

\[
\text{LocHom}_c(T,Q) = \frac{1}{|Q|} \frac{SDev(T,Q)}{\text{Mean}(T,Q)}
\]

\[
\text{EthSco}_c(I,Q) = \frac{\sum_{i=1}^{\|Q\|} \text{GloHet}_k(I_i) \times \text{LocHom}_c(I_i,Q)}{|I|}
\]

\( T \) is a token; \( I \) is a string of tokens \( I_1, I_2, \ldots, I_n \) (e.g. a proposition); \( C \) is a corpus with sections \( c_1, c_2, \ldots, c_N \) representing the EQTs; each \( c_i \) consists of a set of documents \( d_{1,i}, d_{2,i}, \ldots, d_{M,i} \); \( SDev(T,c) \) is the standard deviation of frequency values \( \text{Freq}(T,d_{1,i}), \text{Freq}(T,d_{2,i}), \ldots, \text{Freq}(T,d_{M,i}) \) while \( \text{Mean} \) is the average of the same values. The \( \text{Mean} \) part of the formulae serve to relativize the variance values making small and large tokens numerically comparable.

The relative difference between two ethical profiles is computed by simply summing up the differences between the individual EthSco values.

\[
\text{Diff}(E_P, E'_P) = \sum_{i=1}^{\|Q\|} \frac{\text{ABS}(V_i - V'_i)}{|Q|}
\]

\( E_P \) and \( E'_P \) are ethical profiles both defined on a domain \( Q \) of EQTs: \( \{<Q_1, V_1>, <Q_2, V_2>, \ldots, <Q_{\|Q\|}, V_{\|Q\|}> \} \) and \( \{<Q_1, V'_1>, <Q_2, V'_2>, \ldots, <Q_{\|Q\|}, V'_{\|Q\|}> \} \), respectively.\(^3\) The Diff formula plays a central role in the action matching and agent matching procedures (see fig. 1). Profile matching is computationally cheap, and agent profiles can be computed off-line, ensuring scalability and short response times.

Other EthSco and Diff formulae can be conceived, and many have been considered (see note 2). The present version is the answer to a number of desiderata. EthSco calculation is invariant (everything else being equal to):

- input length (example: "save on oil" has the same ethical profile as "save on oil, save on oil, save on oil")
- token count (example: types "social", "fair", and "opposition" have approximately same GloHet value while they are very different in size: 841ppm, 116ppm, and 47ppm)
- absolute size of EQTs (example: doubling each EQT in \( C \) does not change the EthPro for a fixed \( I \))

\( ^2 \)Our early EthSco formulae did not incorporate the EQT internal distribution (LocHom), leading to an unwanted bias e.g. in cases where a single document had a large number of occurrences of a particular type absent from the other documents in the EQT and the Corpus in general.

\( ^3 \)This Diff formula takes \( E_P \) and \( E'_P \) to be defined for the same \( Q \); this requirement is easily relaxed so that even profiles defined for distinct EQT sets are comparable as long as there is a (preferably substantial) overlap.

- relative size of EQTs (ex: doubling each document in a EQT does not change the LocHom value for a fixed \( I \))
- corpus composition (ex: profiles with distinct EQT composition are directly numerically comparable, given a non-empty intersection).

These invariances are desirable, making corpus maintenance much more flexible. One can add texts to the Corpus continuously, even to selected EQTs only, and still maintain backwards compatibility. Also, the ethical profiles stored in the Actions Database do not have to be refreshed each time a new document - or even a new EQT - is introduced in the Corpus.

**Discourse planning**

When invoked by a client, Freddy presents himself and then continue to produce a prompting question such as "What actions do you take to improve the world?"

The client's answer is scored using EthSco, and the derived ethical profile is used for action matching in A (see fig. 1). This process returns two lists, (i) all actions in A and, (ii) all clients in A, sorted by EthPro proximity. These lists are passed on to the Discourse planning module (essentially an Eliza-style text generator, presented here by an example only). As the generated reply depends directly on the set of actions in A, the system responses change over time as more and more Client submissions are added.

Client: I'm always riding my bike to work

Freddy: Instead of taking the car?

In this case, Freddy found a near-match in A:

"I take my bike to work instead of taking the car"

The cut'n'paste procedure relies on fairly superficial word string comparisons, but with ethical profile matching as a relevance control regime.

I take my bike to work instead of taking the car

I'm always riding my bike to work

"I take" and "I'm always riding" only contain types with low GloHet (interpreted as a license to ignore) while the final phrase "instead .. car" has an EthPro closely matching that of "my bike to work" (interpreted as a license to quote).

Text bits from the Corpus may be used as fillers in the same manner when no adequate match can be found in A.

When Freddy has gathered a few client actions, the exit point is reached where Freddy offers a link to a discussion partner (again based on ethical profile comparisons).

Freddy: Do you wish to chat with another person who is also conscious of her carbon emission footprint?

The user may answer 'yes' or choose to continue the conversation with Freddy.

\( ^4 \)Ethical profiles for agents (represented as sets of actions) can be computed in two ways: either by pooling all tokens produced by an agent calculating a single profile, or by scoring each action separately and then calculate the mean of all profiles. We are currently testing both methods.
EVALUATION
In a preliminary evaluation session, Freddy was tested by 8 adult users all fluent in English. Each user was given three "ethical personalities" picked from the global EQT domain, in turn answering Freddy's questions from each EQT perspective. Freddy's final decisions ("Freddy response" in fig.1) were recorded and later evaluated in a blind review. The procedure was: User $U_i$ receives EFTs $E_1$, $E_2$, and $E_3$. Freddy selects partners $P_1$, $P_2$, $P_3$, respectively. Users $U_i$ through $U_j$ are presented with $E_1$, $E_2$, $E_3$ and $P_1$, $P_2$, $P_3$ in random order and asked to pair them (they are allowed to consult all actions of $P_1$, $P_2$, $P_3$ in the Actions Database). Same procedure is repeated for all users, resulting in 21 $E-P$ pairs for each user, or 168 pairs in total.

A random pairing gets 33.3% correct pairs on average. In contrast, Freddy got 81% correct pairs (136 of 168). In spite of his formal simplicity, Freddy seems to make reasonable decisions within a very sophisticated knowledge domain.

More elaborate test sessions are in preparation, including user ratings of the interaction with the system.

DISCUSSION
In the proposed client classification system, property definitions are replaced by structured data repositories. Whereas lexicographers recommend homogeneous denotation principles, our document collection benefits from including as many independent, heterogeneous text sources as possible, blogs, Twitter, Google search, interest groups - anything but traditional formal explanations. This way the plurality of term uses will be maximally covered.

Morphological and syntactic analysis (lemmatizing, stemming, PoS tagging, parsing) is avoided in the current system. Types like "obey", "obeys", "obeyed", "obedience", "obedient" are treated as independent atoms, as are "I", "me", "my", "we", etc. Most Q/A systems employ at least some linguistic parsing (e.g. [1], [5], [6]) in order to support compositional-semantic decoding in structured knowledge fields such as appointment management ([2]). Information retrieval from web pages ([3]), and tutorial services ([4]).

From a developer's point of view, data-driven methods however have several advantages.
- language independent input analysis procedure
- no need for labor intensive rule writing
- robust to unknown words, misspellings, broken syntax
- the system performance is improving over time, new input being added to the existing database

In many information retrieval systems, semantically empty words like pronouns, copula verbs, and prepositions are ignored. However, we did some experiments showing that even oppositions like "we" and "I", expected to be ethically neutral, actually do contribute to the distinctive powers of Freddy. By way of an example, the $C:we$ ratio is more than four times higher in Veganism (a personal goal) than in Environmentalism (a social goal). Even if vegetarian and environmentalist goals are often overlapping - such as reducing the consumption of animal protein - Freddy is able to discover the subtle ethical difference between "I should eat less meat" and "we should eat less meat". We therefore do not edit the input string before computing its EthPro.

A similar discussion concerns the syntactic structure of the user input. The current Freddy treats the user input as a bag of words with no intrinsic ordering. Most designers of dialogue systems would probably argue that such an architecture loses useful information. This is undoubtedly true for highly controlled input, but Freddy is intended as an informal chat partner and must be prepared for input with erroneous spelling, broken syntax, and even deliberate chat-style reductions making rule-based methods vulnerable. Still some shallow syntax parsing may be rewarding; we are currently looking into this possibility.

The presented algorithm is still in the making, but we do have a first conclusion: People like to chat with Freddy. We suggest that conversation-driven client matching should play an active role in next-generation social recommender systems offering the client an experience of personal service.

REFERENCES
How social relationships affect user similarities

Alan Said
DAI-Labor
Technische Universität Berlin
alan.said@dai-labor.de

Ernesto W. De Luca
DAI-Labor
Technische Universität Berlin
ernesto.deluca@dai-labor.de

Sahin Albayrak
DAI-Labor
Technische Universität Berlin
sahin.albayrak@dai-labor.de

ABSTRACT
In this paper we present an analysis of the social movie recommendation community Filmtipset. Filmtipset is Sweden’s largest movie recommendation community with more than 80,000 users. The website offers movie recommendations based on usage data, but also consists of a social network where users are able to befriend one another. All content is user-generated and there is a multitude of features that make the dataset stand out among other movie recommendation datasets. We evaluate the social graphs’ impact on users similarities in taste in movies, and show that utilizing this relation could be used to improve movie recommendation quality.

ACM Classification Keywords
E.0 Data: General; H.3.3 Information Systems Applications: Information Search and Retrieval; H.4.m Information System Applications: General

General Terms
Analysis, Data

Author Keywords
machine learning, movie recommendations, social networks, collaborative filtering

INTRODUCTION
Movie recommendation websites have been an integral part of the Web for almost as long as the Web has been around, one of the first being The Internet Movie Database\(^1\), actually predating the first web browser\(^6\). In the last couple years movie recommenders have experienced a renaissance with a multitude of new services trying to establish themselves as movie recommenders. One of the reasons movie recommenders have been a hot topic throughout the last couple of years has been the public availability of real-world data. Thanks to services such as MovieLens\(^5\), which makes data available via the GroupLens\(^6\) research team, and Netflix\(^7\), who have made their datasets available as part of the Netflix Prize\(^8\). The amount of research conducted in this domain has been increasing steadily. There is one drawback though, the openly available datasets have all had the same, or a very similar, structure. Most of these datasets provide researchers with two basic relations – a rated user-movie relation consisting of a users’ rating for a particular movie, and a movie-genre relation stating which genre, or genres, a movie belongs to. GroupLens, being one of the exceptions, provides the GroupLens 10M100K dataset\(^9\) which also contains personal tags assigned by users to movies. Filmtipset, on the other hand, not only provides relations well known from other datasets, it also introduces several other features, such as actor ratings, review ratings, and a social graph connecting its users.

The focus of this paper is to evaluate and understand the implications of the user-user relationships found in the social graph in a recommendation scenario. Where most user similarities are derived from the user-movie relations, we use user-user relations instead. For this we use a snapshot of the social graph of the movie recommendation community Filmtipset in order to find how user similarities in movie taste correlate to the social relations between users. We show that friendship does, in fact, affect users taste in movies and that this friendship increases user similarities by nearly 100%.

RELATED WORK
As mentioned in the previous section, there is a vast amount of related work in the field of movie recommendation. The winners of the Netflix Prize, for instance, published a quite provocative paper [8] stating, and showing, that metadata is of little value when it comes to predicting movie ratings. In [2], Amatriain et al., pose that re-rating movies is of signif-

\(^1\)[http://www.imdb.com]
\(^2\)[http://www.jinni.com]
\(^3\)[http://www.moviepilot.com]
\(^4\)[http://www.filmtipset.se]
\(^5\)[http://www.grouplens.org]
\(^6\)[http://www.netflix.com]
\(^7\)[http://www.netflixprize.com]
\(^8\)[http://www.movielens.org/system/files/README_10M100K.html]
icantly higher value than rating new ones. They show how the amount of time that has passed since the original rating affects the users’ new rating, and thus the quality of the recommendations. Other research, not concerning the Netflix Prize, has explored other aspects of movies, such as in [1] where the authors use cultural metadata crawled from comments and reviews in order to boost recommendation results. Ono et al. [7] have instead tried to find the present context of the user in order to give recommendations suiting the users’ current situation.

Guy et al. [5] create a system for recommending items based on a users’ aggregated familiarity network, where relations are extracted from different sources such as co-authorship on wiki pages. The results show that the familiarity network produces better recommendations than classical similarity based approaches. A similar approach is presented in [3] by Bonhard and Sasse.

Golbeck and Hendler [4] are among the very few ones touching on a concept in movie recommendation similar to the one described in this paper. Their approach is based on explicitly defined trust gathered through the FilmTrust\(^\text{10}\) movie recommendation website. FilmTrust asks its users to explicitly assign trust values to their peers, thus stating whose taste to follow and whose not to follow. They conclude that trust does add to the quality of the recommendations. We consider the concept of trust to be related to the explicitly stated friendship relation available in Filmtipset.

DATASET AND EXPERIMENTS
Filmtipset is Sweden’s largest online movie community and has been available to its users since 2000. The service has grown in size and number of features since it started almost ten years ago. At the time of the writing of this paper Filmtipset had more than 80,000 users, 70,000 movies, almost 19 million ratings and more than 10,000 daily visitors. However, its data has never been analyzed and worked on for reasons other than the recommendation service on the website. Therefore we will describe some of the features and attributes of the community not explicitly used in our experiments.

Figure 1 shows a screenshot of the Filmtipset website, similar to most other movie recommendation websites it contains mainly movie related news and links taking the user further into the website.

Figure 2 shows the full set of existing entities and relations in the complete Filmtipset dataset. These features include entities such as user-generated lists, reviews, review ratings, actor comments, etc.\(^\text{11}\). The focus of this paper is the social graph contained in Filmtipset, thus we do not use the full set of entities and relations. The ones used are indicated by the bolder font and darker color used in Figure 2.

The dataset
The dataset used in this paper is a snapshot of a subset of the features available on the website. We focus on the social user-user relations and the user-movie ratings for our experiments and conclusions. This snapshot consists of all ratings posted to the system between April 18, 2000 and September 14, 2009 and all existing friendship relations at the latter date. The exact numbers of entities and relations are shown in Table 1 and Table 2.

<table>
<thead>
<tr>
<th>Entity</th>
<th>No.</th>
<th>Entity</th>
<th>No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movies</td>
<td>67,684</td>
<td>Actors</td>
<td>137,548</td>
</tr>
<tr>
<td>Users</td>
<td>76,505</td>
<td>Directors</td>
<td>28,077</td>
</tr>
<tr>
<td>Comments</td>
<td>1,205,160</td>
<td>Writers</td>
<td>41,830</td>
</tr>
<tr>
<td>Reviews</td>
<td>6,157</td>
<td>Genres</td>
<td>39</td>
</tr>
</tbody>
</table>

Table 1. The entities provided in the dataset.

One of the key features of this website, if not the main, is the social network – users have the possibility to befriend one another in an asymmetric fashion. Asymmetric friendship relations are similar to the follower/following relation on Twitter\(^\text{12}\), meaning that if user \(u_a\) choses to befriend user

\(^\text{10}\)http://trust.mindswap.org/FilmTrust/

\(^\text{11}\)see http://www.filmtipset.se/tailpages.cgi? page=Statistics for some further details

\(^\text{12}\)http://www.twitter.com
Table 2. The relations provided in the dataset.

<table>
<thead>
<tr>
<th>Relation</th>
<th>No.</th>
<th>Relation</th>
<th>No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_a \rightarrow u_b )</td>
<td>29,443</td>
<td>Ratings</td>
<td>18,074,899</td>
</tr>
<tr>
<td>( u_a \leftrightarrow u_b )</td>
<td>53,041</td>
<td>Genre assignments</td>
<td>171,850</td>
</tr>
<tr>
<td>Production roles(^a)</td>
<td>553,981</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^a\) i.e. director, actor, writer

\(u_b\), the latter users’ social graph remains unchanged (in fact, \(u_b\) is not even notified of the event), whereas \(u_a\) will have an explicitly stated friendship relation to \(u_b\).

We take a closer look at the characteristics of the two kinds of friendship relations, i.e. \(u_a \rightarrow u_b\) and \(u_a \leftrightarrow u_b\), asymmetric and symmetric friendships. The asymmetric relations \(u_a \rightarrow u_b\) can be expressed as; \(u_a\) is a fan of \(u_b\), \(u_b\)’s relation graph remains unchanged as the user has not taken a similar action. In the remainder of this paper we will call the relation from \(u_b\) to \(u_a\) friend, as in \(u_b\) is a friend of \(u_a\), the opposite relation will thus be called fan. Furthermore it should be noted that the asymmetric relation \(u_a \rightarrow u_b\) is identical to \(u_b \leftarrow u_a\). In the symmetric case, \(u_a \leftrightarrow u_b\), \(u_a\) and \(u_b\) are both users who have chosen to befriend one another.

![Image](https://via.placeholder.com/150)

Figure 3. The degree distributions of the numbers of friends and fans following a typical power-law distribution.

The degree distribution of the friend and fan relations is shown in Figure 3. We see that both relations follow a typical power-law distributions and are almost identical. This graph contains the symmetric relations as well as the asymmetric ones, as a symmetric relation simply is two asymmetric ones.

![Image](https://via.placeholder.com/150)

Figure 4. Fans versus friends.

In Figure 4 we show the ratios of friends versus fans. We see that the majority of the users that are part of the social graph have less than 50 friends, the average being 4.6 friends and 4.4 fans.

![Image](https://via.placeholder.com/150)

Figure 5. Distribution of number of friends versus number of ratings.

Figure 5 shows the distribution of number of friends versus the number of ratings, we see that if a user has a large number of friends, it is an indication that the same user will often have rated a large number of movies. The opposite rule does not apply though, i.e. a large number of ratings does not imply a large number of social relations.

**Experimental setup**

We evaluate the significance of user-user relations on the overall taste of users, i.e. the number of similar movie every related user pair has seen. In order to do this we convert the ratings in the user-movie matrix from the \{1, 2, 3, 4, 5\} scale that Filmtipset uses to a binary representation. This creates a user-movie relation where users have either seen (1) or not seen (0) a movie. We call this resulting matrix \(C_{\text{all}}\). We then create two additional versions of this matrix; one where all users part of the matrix have at least 5 friends \(C_{\text{um5}}\), and one where all movies and users in the matrix appear at least 5 times \(C_{c5}\). These three matrices are then used to calculate the average Jaccard similarity coefficient for all related user pairs for the user-user relation types declared in Table 2. The Jaccard similarity coefficient is defined as

\[
J(A|B) = \frac{|A \cap B|}{|A \cup B|}
\]

(1)

where \(A\) and \(B\) are the two sets to be compared. We calculate the average Jaccard similarities, \(\bar{J}\), for all related user pairs in the three matrices according to

\[
\bar{J} = \frac{2}{n(n-1)} \sum_{i<j} \left( \frac{|A_i \cap B_j|}{|A_i \cup B_j|} \right)
\]

(2)

where \(n\) is the number of users, and \(i\) and \(j\) two related users. One additional setting is added, namely the pairwise comparison of users who are not part of the social graph, however, this is only done for the original \(C_{\text{all}}\) matrix.

In Figure 4 we show the ratios of friends versus fans. We see that the majority of the users that are part of the social graph have less than 50 friends, the average being 4.6 friends and 4.4 fans.

In Figure 5 we show the distribution of number of friends versus the number of ratings, we see that if a user has a large number of friends, it is an indication that the same user will often have rated a large number of movies. The opposite rule does not apply though, i.e. a large number of ratings does not imply a large number of social relations.

**Experimental setup**

We evaluate the significance of user-user relations on the overall taste of users, i.e. the number of similar movie every related user pair has seen. In order to do this we convert the ratings in the user-movie matrix from the \{1, 2, 3, 4, 5\} scale that Filmtipset uses to a binary representation. This creates a user-movie relation where users have either seen (1) or not seen (0) a movie. We call this resulting matrix \(C_{\text{all}}\). We then create two additional versions of this matrix; one where all users part of the matrix have at least 5 friends \(C_{\text{um5}}\), and one where all movies and users in the matrix appear at least 5 times \(C_{c5}\). These three matrices are then used to calculate the average Jaccard similarity coefficient for all related user pairs for the user-user relation types declared in Table 2. The Jaccard similarity coefficient is defined as

\[
J(A|B) = \frac{|A \cap B|}{|A \cup B|}
\]

(1)

where \(A\) and \(B\) are the two sets to be compared. We calculate the average Jaccard similarities, \(\bar{J}\), for all related user pairs in the three matrices according to

\[
\bar{J} = \frac{2}{n(n-1)} \sum_{i<j} \left( \frac{|A_i \cap B_j|}{|A_i \cup B_j|} \right)
\]

(2)

where \(n\) is the number of users, and \(i\) and \(j\) two related users. One additional setting is added, namely the pairwise comparison of users who are not part of the social graph, however, this is only done for the original \(C_{\text{all}}\) matrix.
The results of these calculations show us the relation-specific average Jaccard similarities between users, e.g. how similar symmetric friends, asymmetric friends and users outside of the social sphere are. The higher this similarity value is, the more movies the users have in common.

RESULTS

The Jaccard similarities for every relation type and every dataset used are presented in Figure 6. The figure depicts the Jaccard similarities for all user pairs in $C_{all}$ ($\forall$), asymmetric friendship pairs in $C_{all}$ ($\leftarrow$), all user pairs where none of the users have any friends in $C_{all}$ ($\emptyset$), all symmetric friendship pairs in $C_{all}$ ($\leftrightarrow$), asymmetric friendship pairs in $C_{um5}$ ($\leftarrow um5$), symmetric friendship pairs in $C_{um5}$ ($\leftrightarrow um5$), asymmetric friendship pairs in $C_{r5}$ ($\leftarrow r5$) and symmetric friendship pairs in $C_{r5}$ ($\leftrightarrow r5$). We see that the three columns depicting the symmetric friendship similarities ($\leftrightarrow$, $\leftrightarrow um5$ and $\leftrightarrow r5$) are almost twice as high (0.18, 0.18 and 0.20) as the average similarity (0.10) of all user pairs ($\forall$).

The higher similarity values in the $C_{um5}$ matrix are expected as we have removed unpopular movies and users who had seen few films, i.e. users having a higher probability of low similarity values to others. The highest similarity value is obtained from the $C_{r5}$ matrix where all users with less than 5 friends have been removed. This indicates that the friendship relationship between users correlates to a similarity in movie taste.

CONCLUSIONS AND FUTURE WORK

We believe that the reason for the higher similarity in the reduced social graph is due to an implicit relation of this graph to the user-movie relations. The results suggest that the reduction of the social graph to a smaller core increases the density of the user-movie matrix, thus, by utilizing the user-user relations in a recommendation scenario one could improve the quality of recommendations. We believe this to be related to the findings of Pilászy and Tikk [8], where movie metadata showed to be of little importance compared to ratings. Our approach, on the other hand, uses information about the users for discovering how social relations can be utilized for recommendation purposes.

These relations provide us with a basis for future explorations of the additional features available in the Filmtipset dataset. We are currently exploring the details of these attributes in the recommendation scenario. Further planned work will focus on extending the user similarities with user-oriented features which, in combination with the obtained results, can further improve the performance of recommendation algorithms.

ACKNOWLEDGMENT

The authors would like to thank the crew behind Filmtipset for their cooperation and assistance.

REFERENCES

8. I. Pilászy and D. Tikk, ‘Recommend new movies: even a few ratings are more valuable than metadata’, in Proc. of the third ACM conf. on Recommender systems, pp. 93–100, New York, New York, USA, (2009). ACM.
User Evaluation Framework of Recommender Systems

Li Chen
Department of Computer Science
Hong Kong Baptist University, Hong Kong
lichen@comp.hkbu.edu.hk

Pearl Pu
Human Computer Interaction Group
Swiss Federal Institute of Technology in Lausanne
pearl.pu@epfl.ch

ABSTRACT
This paper explores the evaluation issues of recommender systems particularly from users’ perspective. We first show results of literature surveys on human psychological decision theory and trust building in online environments. Based on the results, we propose an evaluation framework aimed at assessing a recommender’s practical ability in providing decision support benefits to end-users from various aspects. It includes both accuracy/effort measures and a user-trust model of subjective constructs, and a corresponding sample questionnaire design.

Author Keywords
Recommender systems, user evaluation, adaptive decision theory, trust building, decision accuracy and effort.

ACM Classification
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
Recommender systems emerged as an independent research area since the appearance of papers on “collaborative filtering” in the mid-1990s to resolve the recommendation problem [54]. The automated collaborative filtering (ACF) originated as an information filtering technique that used group opinions to recommend information items to individuals. For instance, the user will be recommended items that people with similar tastes and preferences liked in the past. Various collaborative algorithms based on data mining and machine learning techniques (e.g. K-nearest neighbor, clustering, classifier learning) have been developed to reach the goal. A typical application is MovieLens that predicts the attractiveness of an unseen movie for a given user based on a combination of the rating scores derived from her nearest neighbors [46]. At Amazon.com, the “people who bought this book also bought” was also one example of the commercial adoptions of this technology. Recently, Bonhard et al. showed ways to improve the user-user collaborative filtering techniques by including information on the demographics similarity [8].

In the case that relationship among products is stronger than among customers, content-based recommender methods have been often used to compute the set of items that are similar to what the user has preferred in the past [1]. For example, Pandora, an online music recommender tool, can suggest a sequence of music the user would probably like according to the features (e.g. genre, musician) of ones that she has indicated her preferences on.

Another branch of recommender systems, called preference-based or knowledge-based systems, has been mainly oriented for high-involvement products with well-defined features (such as computers, houses, cars), for which selection a user is willing to spend considerable effort in order to avoid any financial damage [61, 52]. In such systems, a preference model is usually explicitly established for each user.

Researchers have previously indicated the challenges for different types of recommenders. For example, as for the collaborative system, its main limitations are new user problem (i.e. a new user having very few ratings would not be able to get accurate recommendations), new item problem (i.e. until the new item is rated by a substantial number of users, the system would not be able to recommend it), and sparsity (i.e. the number of ratings is very small compared to the number of ratings that need to be predicted) [1]. In order to address these problems, the hybrid recommendation approach combining two or more techniques (the combination of content-based and collaborative filtering) has been increasingly explored [9]. Recently, advanced techniques that involve more types of social resources such as tags and social ties (e.g., friendship or membership) have also emerged in order to improve the similarity accuracy between users or items, and classified into a new branch called social recommender systems [27, 63, 65].

However, few studies have stood from users’ angles to consider their cognitive acceptance of recommendations. Moreover, the question is how to evaluate a recommender in terms of its actual impacts on empowering users to make better decisions, except mathematical algorithm accuracy. In the following, we will first show literature reviews on decision theory from the psychology domain to understand users’ decision making heuristics, given that the recommender is inherently a decision support to assist users in making choices. Furthermore, the user-trust building issues that have been promoted in online environments will be discussed and related to specific research questions to recommenders.
ADAPTIVE DECISION MAKER

The goal of a decision support system is to aid the user in making an informed decision consistent with her objectives. The elicitation of user preferences is fundamental for the recommender system to generate products or services that may interest its users. Most of preference elicitation procedures in recent recommender systems can be classified into two main technologies: implicit preference elicitation which has aimed to infer user preferences according to her demographic data, personality, past navigation and purchase behavior, tags, and so on [38, 8]; and explicit preference elicitation that has emphasized on explicitly asking for the user’s preferences during interaction, such as her rating on an item (in collaborative filtering systems) or stating value functions over item features (in utility-based systems). However, recommender systems, that simply depend on initially obtained user preferences to predict recommendations, may not help the user make an accurate decision.

According to the adaptive decision theory [50], user preferences are inherently adaptive and constructive depending on the current decision task and environment, and hence their initial preferences can be uncertain and erroneous. They may lack the motivation to answer demanding initial elicitation questions prior to any perceived benefits [59], and they may not have the domain knowledge to answer the questions correctly.

As a matter of fact, in the last four decades, the classical decision theory has evolved into two conceptual shifts. One shift is the discovery of adaptive and constructive nature of human decision making. Individuals have several decision strategies at their disposal and when faced with a decision they select a strategy depending on a variety of factors related to the task, the context, and individual differences. Additional studies indicated that individuals often do not possess well-defined preferences on many objects and situations, but construct them in a highly context-dependent fashion during the decision process [62,51].

Another shift has occurred in the field of prescriptive decision making and it is called value-focused thinking [35], different from the traditional attribute-focused thinking. In this approach, once a decision problem is recognized, fundamental and relevant values are first identified to creatively identify possible alternatives and to carefully assess their desirability [10].

Based on the two shifts, researchers in areas of decision theory have identified the following typical phenomena that may occur in a person's adaptive decision process.

Context-dependent preferences. An important implication of the constructive nature of preferences is that decisions and decision processes are highly contingent upon a variety of factors characterizing decision problems. First, choice among options is context (or menu) dependent. The relative value of an option depends not only on the characteristics of that option, but also upon characteristics of other options in the choice set. For example, the relative attractiveness of $x$ compared to $y$ often depends on the presence or absence of a third option $z$ [62]. Second, preference among options also depends upon how the valuation question is asked. Strategically equivalent methods for eliciting preferences can lead to systematically different preference orderings. Third, choice among options depends upon how the choice set is represented (framed) or displayed. Finally, the process used to make a choice depends on the complexity of the decision tasks: the use of simple decision heuristics increases with task complexity [51].

Four decision metagoals. Evidence from behavioral studies indicates four main metagoals driving human decision making. Although individuals clearly aim at maximizing the accuracy of their decisions, they are often willing to tradeoff accuracy to reduce cognitive effort. Also, because of their social and emotional nature, when making a decision, people try to minimize/maximize negative/positive emotions and maximize the ease of justifying a decision [5]. When faced with a decision, people make critical assessments of the four metagoals contingent on the decision task (e.g. number of alternatives) and the decision environment (e.g. how information is presented to the decision maker). Especially in unfamiliar and complex decision conditions, decision makers reassess the metagoals and switch from one strategy to another as they learn more about the task structure and the environment during the course of decision making [50].

Anchoring effect. Researchers also suggested that people use an anchor-and-adjust strategy to solve a variety of estimation problems. For example, when asked questions about information that people do not know, they may spontaneously anchor on information that comes to mind and adjust their responses in a direction that seems appropriate [34]. This heuristic is helpful, but the final estimate might be biased toward the initial anchor value [19].

Tradeoff avoidance. Decision problems often involve conflict among values, because no one option is best on all attributes of values, and conflict has long been recognized as a major source of decision difficulty [56]. Thus, many researchers argued that making tradeoffs between more of one thing and less of another is a crucial aspect of high-quality and rational decision making [21]. However, decision makers often avoid explicit tradeoffs, relying instead on an array of non-compensatory decision strategies [49]. The explanation for tradeoff avoidance is that tradeoffs can be difficult for emotional as well as cognitive reasons [31, 40].

Means objectives. According to value-focused thinking (VFT), the decision maker should qualitatively distinguish between fundamental and means objectives. Fundamental objectives should reflect what the decision maker really wants to accomplish with a decision, while means objectives simply help to achieve other objectives [36].
However, inadequate elicitation questions can easily circumscribe a user in thinking about means objectives rather than fundamental objectives. For example, a traveler lives near Geneva and wants to be in Malaga by 3:00 pm (her fundamental objective), but if she was asked to state departure time, she would have to formulate an objective (i.e., departure at 10:00 am), even though there is a direct flight that leaves at 2:00 pm.

Therefore, as suggested in [51], metaphorically speaking, preference elicitation is best viewed as architecture (building a set of values) rather than archeology (uncovering existing values). In order to avoid human decision biases, preference elicitation tools must attempt to quickly collect as much preference data as possible so that users can begin working towards their goals. Furthermore, they must also be able to resolve potential conflicting preferences, discover hidden preferences, and make reasonable decisions about tradeoffs with competing user goals.

Unfortunately, most of current recommender system designs did not recognize the importance of these implications. In order to help the user make an accurate and confident decision, we have been mainly engaged to realize a decision aid that can embody all of the requirements. In addition, by means of user experience research, we have attempted to derive more useful principles for the development of an intelligent and adaptive preference-based recommender system.

TRUST BUILDING IN ONLINE ENVIRONMENTS

The second challenge is about how to build user trust in recommender systems. Less attention has been paid in related work to evaluating and improving the recommender system from the aspect of users’ subjective attitudes. Among the many factors, the perception of the recommender’s trustworthiness would be most prominent as it facilitates long-term relationship and encourages potential repeat interactions and purchases [22, 17].

Trust has been in nature regarded as a key factor to the success of e-commerce [23]. Due to the lack of face-to-face interaction with consumers in online environments, users’ actions undertake a higher degree of uncertainty and risk than in traditional settings. As a result, trust is indeed difficult to build and easy to lose with the virtual store, which has impeded customers from actively participating in e-commerce environments [33].

The definition of trust has varied from study to study. The most frequently cited definition of trust in various contexts is the “willingness to be vulnerable” proposed by Mayer et al. [42]. Adapting from this definition, Chopra and Wallace defined trust in the electronic environment as the “willingness to rely on a specific other, based on confidence that one’s trust will lead to positive outcomes.” [15] More specifically, consumer trust in online shopping was defined as “the willingness of a consumer to expose himself/herself to the possibility of loss during an Internet shopping transaction, based on the expectation that the merchant will engage in generally acceptable practices, and will be able to deliver the promised products or services.” [39]

As these definitions indicate, consumer trust is essentially leading to behavioral intentions [24], referred as “trusting intentions” by McKnight et al. [45]. Consistent with the Theory of Planned Behavior [2], consumer trust (as a belief) will influence customer intentions. Empirical studies have shown that trust in an e-commerce website increases customer intention to purchase a product from the website, as well as intention to return to it for future use. Other potential trusting intentions include providing personal information (email, phone number and credit card number) and continuing to transact with the website [26].

Many researchers have also experimentally investigated the antecedents of on-line trust. For example, Pavlou and Chellappa explained how perceived privacy and perceived security promote trust in e-commerce transactions [48]. De Ruyter et al. examined the impact of organizational reputation, relative advantage and perceived risk on trust in e-service and customer behavior intentions [55]. Jarvenpaa et al. validated that the perceived size of an Internet store and its perceived reputation are positively related to consumers’ initial trust in the store [33].

The effect of experience with website interface on trust formation has been also investigated based on the Technology Acceptance Model (TAM) [16,68]. TAM has long been considered a robust framework for understanding how users develop attributes towards technology and when they decide to adopt it. It posits that intention to voluntarily accept and use a new information technology (IT) is determined by two beliefs: the perceives usefulness of using the new IT, and the perceived ease of use of the new IT. According to TAM, Koufaris and Hampton-Sosa established a trust model and demonstrated that both the perceived usefulness and the perceived ease of use of the website are positively associated with customer trust in the online company and customer’ intentions to purchase and return [37]. Gefen et al. expanded TAM to include a familiarity and trust aspect of e-commerce adoption, and found that repeat customers’ purchase intentions were influenced by both their trust in the e-vendor and their perceived usefulness of the website, whereas potential customers were only influenced by their trust [25]. Hassanein and Head identified the positive influence of social presence on customers’ perceived usefulness of an e-commerce website and their trust in the online vendor [29].

In the domain of recommender systems, trust value has been also noticed but it has been mainly used to empower the prediction of user interests, especially for the collaborative filtering (CF) systems [32]. For instance, O’Donovan and Smyth have proposed a method to incorporate the trustworthiness of partners into the standard computation process in CF frameworks in order to increase the predictive accuracy of recommendations [47]. Similarly,
Massa and Bhattacharjee developed a trust-aware technique taking into account the “web of trust” provided by each user to estimate the relevance of users’ tastes in addition to similarity measure [41]. Few literatures have highlighted the importance of user trust in recommender systems and proposed effective techniques to achieve it. The studies done by Swearingen and Sinha showed the positive role of transparency, familiarity of the recommended items and the process for receiving recommendations in trust achievement [60]. Zimmerman and Kurapati described a method of exposing the reflective history in user interface to increase user trust in TV recommender [66].

However, the limitations are that there is still lack of in-depth investigations of the concrete system design features that could be developed to promote user trust, and lack of empirical studies to measure real-users’ trust formation and the influential constructs that could be most contributive to users’ behavioral intentions in a recommender system.

Considering these limitations, our main objective is to explore the crucial antecedents of trustworthiness for recommender systems and their exact nature in providing benefits to users. Concretely, driven by the above decision theory findings and trust issues, we have built an evaluation framework aimed at including all of crucial standards to assess a recommender’s true ability.

**EVALUATION FRAMEWORK**

As a matter of fact, identifying the appropriate criteria for evaluating the true benefits of a recommender system is a challenging issue. Most of related user studies purely focused on users’ objective performance such as their interaction cycles and task completion time [43], less on decision accuracy that the user can eventually achieve, and subjective effort that the user cognitively perceived in processing information. Moreover, as mentioned above, the consumer trust should be also included as a key standard, such as whether the recommender could significantly help to increase users’ competence-inspired trust and furthermore their behavioral intention to purchase a product or intention to return to it for repeated uses.

**Decision Accuracy and Decision Effort**

According to [50], two key considerations underlying a user’s decision strategy selection are: the accuracy of a strategy in yielding a “good” decision, and the “cognitive effort” required of a strategy in making a decision. All else being equal, decision makers prefer more accurate choices and less effortful choices. Unfortunately, strategies yielding more accurate choices are often more effortful (such as weighted additive rule), and easy strategies can sometimes yield lower levels of accuracy (e.g. elimination-by-aspects). Therefore, they view strategy selection to be the result of a compromise between the desire to make the most correct decision and the desire to minimize effort. Typically, when alternatives are numerous and difficult to compare, like when the complexity of the decision environment is high, decision makers are usually willing to settle for imperfect accuracy of their decisions in return for a reduction in effort. The observation is well supported by [6, 57] and consistent with the idea of bounded rationality [58].

A standard assumption in past research on decision support systems is that decision makers who are provided with decisions aids that have adequate information processing capabilities will use these tools to analyze problems in greater depth and, as a result, make better decisions [30, 28]. However, empirical studies also showed that because feedback on effort expenditure tends to be immediate while feedback on accuracy is subject to delay and ambiguity, the use of decision aids does not necessarily enhance decision making quality, but merely leads individuals to reduce effort [18, 4].

Given this mixed evidence, it cannot be assumed that the use of interaction decision aids will definitely enhance users’ decision quality. Thus, an open question to recommender systems is that whether they could enable users to reach the optimal level of accuracy under the acceptable amount of effort users are willing to exert during their interaction with the system. In the following, we introduce our accuracy-effort measurement model, derived from the ACE (Accuracy, Confidence, Effort) framework that we have previously built for preference-based product recommenders [67]. Decision accuracy and decision effort are respectively evaluated from both objective and subjective dimensions and their tradeoff relationship is also included as shown in Figure 1.

**Objective and Perceived Decision Accuracy**

In related work, decision accuracy has been measured adaptive to different experimental situations or purposes. In Payne et al.’s simulations, the accuracy of a particular heuristic strategy was defined by comparing its produced choice against the standard of a normative model like the weighted additive rule (WADD) [50]. The performance measures of precision and recall have been commonly applied to test an information retrieval system’s accuracy based on a set of ground truths (previously collected items that are relevant the user's information need) [7]. In the condition of user experience researches, Haubl and Trifts suggested three indicators of a user’s decision quality: increased probability of a non-dominated alternative selected for purchase, reduced probability of switching to another alternative after making the initial purchase decision, and a higher degree of confidence in purchase decisions [28]. In our case, we considered two facets: objective decision accuracy and perceived accuracy.
**Objective Decision Accuracy.** It is defined as the quantitative accuracy a user can eventually achieve by using the assigned decision system to make a choice. More specifically, it can be measured by the fraction of participants whose final option found with the decision tool agrees with the target option that they find after reviewing all available options in an offline setting. This procedure is known as the switching task. Switching refers to whether a user switches to another choice of product after reviewing all products instead of standing by the choice made with the tool. In our previous experiments [11,53], the “switching” task was supported by both sorting and comparison facilities. Subjects were encouraged to switch whenever they saw an alternative they preferred over their initial choice.

A lower switching fraction, thus, means that the decision system allows higher decision accuracy since most users are able to find their best choice with it. On the contrary, a higher switching fraction implies that the system is not very capable of guiding users to obtain what they truly want. For expensive products, such inaccurate tools may cause both financial damage and emotional burden to a decision maker.

**Perceived Accuracy.** Besides objective accuracy, it is also valuable to measure the degree of accuracy users subjectively perceived while using the system, which is also called decision confidence in some literatures [52]. The confidence judgment is important since it would be likely associated with users’ competence perception of the system or even their intention to purchase the chosen product. The variable is concretely assessed either by asking subjects to express any opinions on the interface or directly requiring them to rate a statement like “I am confident that the product I just ‘purchased’ is really the best choice for me” on a Likert scale ranging from “strongly disagree” to “strongly agree”.

**Objective and Perceived Decision Effort**

According to the accuracy–effort framework [50], another important criterion of evaluating a decision system’s benefit is the amount of decision effort users expend to make their choice. So far, the most common measure appearing in related literatures is the number of interaction cycles or task time that the user actually took while using the tool to reach an option that she believes to be the target option. For example, session length (the number of recommendation cycles) was regarded as an importance factor of distinguishing the Dynamic Critiquing system with its compared work like FindMe interfaces [43]. In our model, we not only care about how much objective effort users actually consumed, but also their perceived cognitive effort, which we hope would indicate the amount of subjective effort people exert.

**Objective Effort.** The objective effort is concretely reflected by two dimensions: the task completion time and the interaction effort. The interaction effort was either simply defined as the total interaction cycles users were involved, or divided into more detailed constructs if they were necessary to indicate an average participant’s effort distribution. For instance, in an online shopping setting, the interaction effort may be consumed in browsing alternatives, specifying filtering criteria, viewing products’ detailed information, putting multiple products into a consideration set, and so on. Such effort components were also referred to Elementary Information Processes (EIPs) for a decision strategy’s effort decomposition [50,64].

**Perceived Cognitive Effort.** Cognitive decision effort indicates the psychological cost of processing information. It represents the ease with which the subject can perform the task of obtaining and processing the relevant information in order to enable her to arrive at her decision. Normally, two or more scale items (e.g. “I easily found the information I was looking for”) can be used to measure the construct perceived effort. The respondents were told to mark each of items on a Likert scale ranging from “Strongly Disagree” to “Strongly Agree”.

**Trust Model for Recommender Systems**

As indicated before, trust is seen as a long term relationship between a user and the organization that the recommender system represents. Therefore, trust issues are critical to study especially for recommender systems used in e-commerce where the traditional salesperson, and subsequent relationship, is replaced by a product recommender agent. Studies showed that customer trust is positively associated with customers' intention to transact, purchase a product, and return to the website [33]. These results have mainly been derived from online shops' ability to ensure security, privacy and reputation, i.e., the integrity and benevolence aspects of trust formation, and less from a system’s competence such as a recommender system’s ability to explain its result.

These open issues led us to develop a trust model for building user trust in recommender systems, especially focusing on the role of competence constructs. The term “trust” is theoretically defined by a combination of trusting beliefs and trusting intentions, in accordance with the Theory of Planned Behavior (TPB) asserting that behavior is influenced by behavior intention and that intention is determined by attitudes and beliefs [2]. So we first introduce TPB and Technology Acceptance Model, based on which our trust model has been established.

**Theory of Planned Behavior**

In psychology, the theory of planned behavior (TPB) is a theory about the link between attitudes and behavior. It was proposed by Icek Ajzen as an extension of the theory of reasoned action (TRA) [20,2]. It is one of the most predictive persuasion theories. It has been applied to studies of the relations among beliefs, attitudes, behavioral intentions and behaviors in various fields such as advertising, public relations, campaigns, healthcare, etc.

TPB posits that individual behavior is driven by behavioral intentions where behavioral intentions are a function of an individual’s attitude toward the behavior, the subjective norms surrounding the performance of the behavior, and the
individual’s perception of the ease with which the behavior can be performed (behavioral control) (see Figure 2).

Attitude toward the behavior is defined as the individual’s positive or negative feeling about performing a behavior. It is determined through an assessment of one’s beliefs regarding the consequences arising from a behavior and an evaluation of the desirability of these consequences. Subjective norm is defined as an individual’s perception of whether people think their significant others wanted them to perform the behavior. The contribution of the opinion of any given referent is weighted by the motivation that an individual has to comply with the wishes of that referent. Behavioral control is defined as one’s perception of the difficulty of performing a behavior. TPB views the control that people have over their behavior as lying on a continuum from behaviors that are easily performed to those requiring considerable effort, resources, etc.

**System Design Features.** The system features basically deal with all design aspects of a recommender system that will probably contribute to the promotion of its overall competence perceptions. Concretely, they include the interface display techniques such as the explanation-based interface to give system transparency, the recommendation quality to reflect users’ perception of the recommender algorithm’s accuracy and the user-system interaction models like the allowed degree of user control.

**Competence Constructs.** It is widely accepted that competence, benevolence and integrity explain a major portion of a trustee’s trustworthiness [23]. Among them, we believe that the competence perception would be most reflective of system design qualities of the recommender. Based on TAM and related works [16,37], we include typical constructs of perceived ease of use, perceived usefulness, and an additional capacity assessment enjoyment. Moreover the two subjective measurements of decision accuracy and decision effort are also involved to represent the system’s decision support quality.

**Trustworthiness.** The “trustworthiness” (or called credibility) [42] is the main positive influence on trusting intentions [23,44]. In our model, it is generally assessed by two major constructs: trust in recommendations and users’ overall satisfactory degree with the recommender interface.

**Trusting Intentions.** Trusting intention is the extent to which the user is willing to depend on the technical party in a given situation [45]. We include in our model the intention to purchase (i.e. purchase a product from the website where the recommender is found) and the intention to return (i.e. return to the recommender system for more products information), as most of e-commerce based trust models emphasize. In addition, we added the intention to save effort to address whether the recommender system could allow its users to benefit from the built trust. That is, whether upon establishing a certain trust level with the recommender at the first visit, users will more readily accept the recommended items, rather than exerting extra effort to process all information themselves, once returning to use it.

**Figure 2. The model of Theory of Planned Behavior [2].**

Technology acceptance model is another influential extension of Ajzen and Fishbein’s theory of reasoned action (TRA) [20]. Some online trust models were built based on it especially when they examined user experience with Web technologies.

It was developed by Fred Davis and Richard Bagozzi to model how users come to accept and use a technology [3,16]. The model suggests that when users are presented with a new software package, a number of factors (replacing many of TRA’s attitude measures) influence their decision about how and when they will use it.

TAM posits that perceived usefulness and perceived ease of use determine an individual’s intention to use a system, with intention to use serving as a mediator of actual system use. Perceived usefulness is also seen as being directly impacted by perceived ease of use. Formally, perceived usefulness (PU) was defined as “the degree to which a person believes that using a particular system would enhance his or her job performance”, and perceived ease-of-use (PEOU) is “the degree to which a person believes that using a particular system would be free from effort” [16].

**Trust Model**

Inspired by the two theories, our trust model consists of four main components specific to recommender systems: system design features, competence constructs, trustworthiness of the system and trusting intentions (see Figure 3 of the model)

**Figure 3. The user-trust model for recommender systems.**

Therefore, as shown in Figure 3, all subjective variables are grouped into four categories. During the user evaluation of a system, in addition to analyzing each single variable, it will be also interesting to identify the relationships between different variables through correlation analysis. For instance, how the perceptions of system-design features are associated
with specific constructs of competence assessments, and how the competence constructs influence trust promotions, which would furthermore affect trusting intentions.

Table 1 lists all of the questions that can be adopted to measure these subjective variables. Most of them came from existing literatures where they have been repeatedly shown to exhibit strong content validity and reliability. Each question is required to respond on a 5-point Likert Scale ranging from "strongly disagree" to "strongly agree".

**Table 1. Questions to measure subjective constructs in our trust model.**

<table>
<thead>
<tr>
<th>Measured variable</th>
<th>Question responded on a 5-point Likert scale from “strongly disagree” to “strongly agree”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subjective perceptions</td>
<td></td>
</tr>
<tr>
<td>Transparency</td>
<td>I understand why the products were returned through the explanations in the interface.</td>
</tr>
<tr>
<td>Recommendation quality</td>
<td>This interface gave me some really good recommendations.</td>
</tr>
<tr>
<td>User control</td>
<td>I felt in control of specifying and changing my preferences in this interface.</td>
</tr>
<tr>
<td>Overall competence perceptions</td>
<td></td>
</tr>
<tr>
<td>Perceived ease of use</td>
<td>I find this interface easy to use.</td>
</tr>
<tr>
<td>Perceived usefulness</td>
<td>This interface is competent to help me effectively find products I really like.</td>
</tr>
<tr>
<td></td>
<td>I find this interface is useful to improve my “shopping” performance.</td>
</tr>
<tr>
<td>Enjoyment</td>
<td>I found my visit to this interface enjoyable.</td>
</tr>
<tr>
<td>Decision confidence</td>
<td>I am confident that the product I just “purchased” is really the best choice for me.</td>
</tr>
<tr>
<td>Perceived effort</td>
<td>I easily found the information I was looking for.</td>
</tr>
<tr>
<td></td>
<td>Looking for a product using this interface required too much effort (reverse scale).</td>
</tr>
<tr>
<td>Trustworthiness</td>
<td></td>
</tr>
<tr>
<td>Trust in recommendations</td>
<td>I feel that this interface is trustworthy.</td>
</tr>
<tr>
<td></td>
<td>I trust the recommended products since they were consistent with my preferences.</td>
</tr>
<tr>
<td>Satisfaction</td>
<td>My overall satisfaction with the interface is high.</td>
</tr>
<tr>
<td>Trusting intentions</td>
<td></td>
</tr>
<tr>
<td>Intention to purchase</td>
<td>I would purchase the product I just chose if given the opportunity.</td>
</tr>
<tr>
<td>Intention to return</td>
<td>If I had to search for a product online in the future and an interface like this was available, I would be very likely to use it.</td>
</tr>
<tr>
<td></td>
<td>I don’t like this interface, so I would not use it again (reverse scale).</td>
</tr>
<tr>
<td>Intention to save effort in next visit</td>
<td>If I had a chance to use this interface again, I would likely make my choice more quickly.</td>
</tr>
</tbody>
</table>

**CONCLUSION**

Thus, as a summary, our evaluation framework is mainly composed of two important components: the accuracy-effort measures and the user-trust model. The objective accuracy and effort are respectively measured by observing users’ switching rate, recording their interaction effort and time consumed to accomplish their search tasks. Regarding subjective measures such as perceived accuracy, perceived effort and trust-related constructs, a post-study questionnaire is designed to ask for users’ subjective opinions and comments after they finished their decision tasks with the assigned recommender system.

We have previously conducted a series of user studies with the goal of consolidating the evaluation framework. For example, *example-critiquing systems*, that support users to provide explicit feedbacks to recommendations, have been mainly tested regarding users’ decision accuracy and decision effort [52, 53, 11]. The user evaluations found that the system can significantly improve decision accuracy compared to non-critiquing systems, while demanding similar level of effort. Another user study on *organization-based explanation interface* identified the explanation’s positive role in increasing users’ competence perception and return intention as focused in the trust model [12]. The user study on *hybrid critiquing* interface (integrated with system-suggested critiques) took both of users’ decision performance and subjective perceptions into consideration [13, 14].

Given the importance of these evaluation criteria as implied by researches on adaptive decision theory and online-trust building, we believe that this evaluation framework will be useful and scalable to the evaluation of recommender systems in a broad domain including recent social recommender systems [27]. In fact, we have started to extend the subjective constructs with more aspects, such as diversity, novelty, attractiveness, and test their practicability in domains of public tastes (e.g., movie, music). In the future, we will continually validate and refine the framework in different system scenarios in order to generalize its applicable value.

**REFERENCES**


A Step toward Personalized Social Geotagging

A. Di Napoli, F. Gasparetti, A. Micarelli, G. Sansonetti
Department of Computer Science and Automation
Artificial Intelligence Laboratory
Roma Tre University
{alfredo.dinapoli,gaspare,micarel,gsansone}@dia.uniroma3.it

ABSTRACT
In this paper, we propose a framework for personalized access to location-aware services. The system relies on the integration of a Geographic Information Retrieval module with a User Modeling component. We describe an application into a specific field, but the platform is easily usable in different contexts. Currently, the framework has been extending to the mobile domain.

Author Keywords
Geotagging, User Modeling, Ubiquitous Computing

ACM Classification Keywords
H.3.5 Online Information Services: Web-Based Services

INTRODUCTION
Geotagging is a practice developed with Web 2.0. It is the operation of adding informative metadata to photos, videos, podcasts, websites or RSS feeds. Through these data, it is possible to obtain the geographic identification of the file. Nowadays, world-mapping is being done by powerful tools such as Google Street View, so Geotagging is a very common practice for both uploads and downloads, thanks to the convergence between 3g network technology and smartphones, which allows users to obtain and supply information in real-time about the places they are passing through.

In our opinion, some factors do not allow users to fully exploit the potential benefits such services are able to offer. First of all, when the user looks for information about the places she is passing through, she is often overwhelmed with tens of references among which she must find the one actually searched for. Such an operation is made even more complex by the necessarily small size of the screen of mobile devices. An other drawback, which can bother some users, is that the proposed references are usually ranked only partially on a location-based measure. Both of them, in fact, are willing to cover more distance in order to obtain exactly what they are looking for. A possible solution would be representing the short- and long-term users’ needs, in such a manner to fit the suggestions offered by the system.

The goal of the platform we propose here is to provide all users with satisfactory solutions. Specifically, the system we are working on leverages a modeling component which allows the device to provide the user with the requested information, successfully ranked according to her needs.

Moreover, this modeling component allows the system to suggest recommendations without an explicit request of the user. Another feature of the proposed platform is the ease whereby it can be extended to different domains. Later on, we describe an instance in the field of beaches, but the design choices are such that it can be easily extended to different fields.

The system has been implemented through the open-source framework Django, entirely written in Python language. Now the platform runs on desktop, but we are working on its implementation on mobile devices.

The rest of the paper is organized as follows. In the next section, the proposed approach is illustrated. We then discuss the adaptivity of the framework. In the last sections, we review related work and draw our conclusions on what has been done and what is still to be done for the final system.

THE PROPOSED APPROACH
The main idea behind the system is to supply users with a web application in perfect Web 2.0 style, whereby it is possible to share ideas, advises and opinions on itineraries, sites, hotels, beaches, campings, restaurants and much more. The web application we are talking about manages a geographic map freely taggable by registered users. Members can ac-
As we will see, the design and implementation choices allow the system to be used in different contexts. Figure 1 shows the interface of the instance of the framework in the field of beaches. Tags are denoted by small suns.

Several useful information can be associated to each tag, among which the user who posted the tag, the type of description, the text of the description, the user rating, one or more photos and/or videos.

The GeoDjango module of Django has been used to extend the Object-Relational Mapper that allows the execution of queries on spatial databases. Moreover, a software model has been created to manage the descriptions posted by users. Such a model allows everyone to choose any point on javascript maps, initially used by Google Maps, and to tag that point. This information is saved on the server side as geographic points in the spatial database. The integration with the GeoIP module allows to localize the user’s location through a correspondence between IP address and geographic coordinates.

In order to populate the platform a Geographic Information Retrieval library has been realized. The implemented methods are the following ones:

1. A Geo-Retrieval module has been structured to satisfy the following workflow:
   a. some keywords concerning the specific field are extracted;
   b. these keywords are given in input to the Google server;
   c. the results of the query are then analyzed and filtered;
   d. each found KML
   e. each file can contain different geographic sites;

2. The second method for the initial population of the system is to exploit the public web services to already geo-tagged data. For example, a plugin of the proposed framework allows one to interface Wikipedia API. A Wikipedia page (see Fig. 2) contains descriptions of beaches already posted by users. Most pages include geographic coordinates, so it is possible to exploit them to populate the system. Geographic coordinates can be expressed in different formats, among which decimal degrees and degrees:minutes:seconds. The former is useful for KML files and parsing. However, most websites (among which Wikipedia) employ the latter: in that case a conversion is needed.

3. Eventually, a focused crawler has been realized, that is a crawler for downloading the web pages pertaining to the specific topic [5]. This tool allows one to populate the framework, once the field has been defined.

ADAPTIVITY

One distinctive feature of the proposed system is the adaptivity provided by user modeling. Collecting information about individual users is essential for a system to provide the adaptation effect, that is to behave differently for different users. For instance, when the user is interested in strong-wind beaches for surfing, the system can adaptively select and prioritize the most relevant items and recommend them next time she logs in.

Users’ interests as well as users’ knowledge are the most important features to be modeled in Adaptive Hypermedia systems [2]. In our framework implicit feedback techniques [4] are employed to collect usage data related to user interests. These data are analyzed to build the user profile employed during the recommendation process.

In the chosen domain, the usage data that are possible to collect are the following:

- **Search history**: the set of keywords submitted by the user for querying the search engine;

3KML is the acronym of Keyhole Markup Language, that is a programming language similar to XML, used by Google to represent geographic sites in Google Earth.
**Click history**: the countries of the items chosen by the user;

**Tag history**: the tags of the items visited or rated by the user;

**Navigation history**: the items that have been visited or rated by the user.

It is possible to generalize the current domain-dependent data to any kind of retrieval system that provides geospatial metadata, which allows users to navigate through this kind of information. These usage data correspond to the user profile. The system is also able to collect relevance feedback data in the form of item ratings. This is the only feature that allows users to submit explicit feedback in order to be able to learn their needs, interests, and preferences. Aside from requiring additional time during the seeking processes, explicit feedback places a cognitive burden on users and the obtained benefits are not always clear (see, for example, [6]), therefore the effectiveness of explicit techniques may be limited.

Once the data have been collected, they must be included in the recommendation process. The interpretation of these data streams is necessary for the system to tailor itself to each individual’s interests. In traditional IR systems, usage data are employed by adjusting the weights of terms in the original query, or by using those data to add words to one query. Relevance feedback is often implemented through the Rocchio Algorithm [1].

Recommender systems relying on content-based retrieval systems suffer from the popular vocabulary problem [3]. Vocabulary variety can lead to serious problems during interactions. Different terms may be used to refer to the same object. When users submit queries, they should predict and use the same vocabulary employed when pages have been authored. For this reason, a traditional content-based approach may fail to retrieve relevant information.

Fortunately, in our domain items are also represented by means of tags. Besides traditional geo tags (e.g., geospatial metadata), the user is able to select tags in order to represent features of objects. The vocabulary employed to define the tag set is usually limited and users are often forced to choose one or more tags from this set. Tag history is one of the profile elements that better represent the users’ interests. It is collected by analyzing the tags of the items visited or rated by the user, and is also extended to the tags chosen by the user during the search.

Our first approach is to retrieve a subset of relevant items by means of a traditional location-based metric, namely items located closer to the user are judged better. The second step is to rank these items according to a metric that weights how many tags of one item are in common with the user profile. In the current prototype, a Jaccard similarity coefficient is used to assess this measure:

\[
\text{similarity}(\text{TAG}_p, \text{TAG}_i) = \frac{\text{TAG}_p \cap \text{TAG}_i}{\text{TAG}_p \cup \text{TAG}_i}
\]

where \(\text{TAG}_p\) is the set of tags included in the user profile and \(\text{TAG}_i\) is the set of items associated with the item \(i\).

The other histories are exploited in order to perform proactive recommendation. Whenever users log in, the system suggests a set of recommendations related to the current interests (see Fig. 1). If the user logs in through a mobile device that is able to send the current geographic coordinates (e.g., latitude and longitude), this information is used to retrieve items in the user neighborhood by means of a traditional location-based metric. The same personalization process previously described is employed to re-rank the current retrieved set according to the user’s interests.

The user has the chance to log in via a traditional browser, where data about geographic coordinates are not available or it is not possible to easily transfer these data to the web servers. In this case, the histories related to searches, clicks and past navigations are exploited to retrieve a set if items to be suggested in the graphical UI. In particular, the set of keywords used to query the system in the previous interactions are used to retrieve a first set of results. The matching is based on a IR system with a cosine similarity measure [11]. This result set is then ranked according to the most visited countries and items visited or rated by the user. Basically, the items located in the countries or in the neighborhood of previous items visited by the user are promoted in the list.

This straightforward approach is able to extend the simple location-based approach employed in traditional map services with geotagging support to include elements related to the current users’ interests and preferences.

**RELATED WORK**

Currently, some software applications already allow users to receive information and suggestions concerning the locations from where they connect. Plazes is a website where users can realize where other users are and what they are doing. In this way, it is possible to share information and to post next plans. These operations can be done on the website http://plazes.com, or through the cell phone. Users can also link to groups to keep in touch with friends and Plazes. The general idea behind this service is to allow everyone to monitor places and people throughout their lives.

The website http://google sightseeing.com is a portal where members can publish information and comments on places of some interest. Specifically, descriptions concern artistic or funny places. Members can also post itineraries or photos by publishing KML files.

Google has just announced that location will be “a first class component of all the products” the company develops. The new Google mobile homepage will shortly report the voice

---

4http://plazes.com
What’s Nearby? 5 to allow users to receive personalized search suggestions based on the location. This way, if the user will search for a specific product, the first results will be related to stores where the product is available and nearest to the user’s location.

Moreover, the same homepage will include the voice Near Me Now 6 to allow everyone to obtain information concerning ATMs, restaurants, bars. The new version of Google mobile maps for Android 7 will include this technology.

Another project promoted by Google concerns the 2D mobile barcode format as a camera-friendly tool to find out local businesses and save money in nearby stores 8. Google has already sent 100,000 mobile barcodes to all American stores. The idea is to show the mobile barcodes in shop windows. Users might be interested in some item, take a picture of the barcode through their cameraphone, and quickly receive info, ratings and coupon.

CONCLUSIONS
In this paper, we have presented a software application which allows personalized access to information available on the web, concerning places and services nearby to the user’s current location. In perfect Web 2.0 style, such an application allows everyone to contribute by entering new places or by adding ratings, comments, photos and videos.

The impressions of the first users are satisfactory. Notwithstanding, a deeper evaluation is needed, for which an opportune test-bed must be defined.

The system is still under development. Among other issues, we are extending portal applications to mobile users on iPhone and Android platforms.

REFERENCES


Relationship Aggregation for Enhancing Enterprise Recommendations

Ido Guy
IBM Haifa Research Lab
Mt. Carmel, Haifa
31905, Israel
ido@il.ibm.com

ABSTRACT
As social media and Web 2.0 sites are becoming ubiquitous, aggregation of social network information becomes valuable, building a more comprehensive and rich social graph. In this position paper we summarize our use of social relationship aggregation for recommendation of people and content within enterprise social software and draw future directions for recommender systems in the social media domain.

Author Keywords
Aggregation, social media, enterprise recommendation

ACM Classification Keywords
H.5.3 Information Interfaces and Presentation: Group and Organization Interfaces – Collaborative computing; H.3.3 Information Search and Retrieval – Information filtering.

General Terms
Algorithms, Design, Experimentation, Human Factors, Measurement

INTRODUCTION
With the evolution of the social web (Web 2.0), people are leaving traces of relationships everywhere. From explicit connections in social network sites (SNSs), such as Facebook and LinkedIn, to more implicit marks, such as co-editing a page in Wikipedia, commenting on a blog in WordPress, or using the same tag in Flickr. Following their counterparts on the Web, social media sites have emerged within large organizations, such as IBM, HP, and Microsoft. These enterprise social media sites also unlock a lot of valuable information about relationships between employees.

SOCIAL NETWORK AGGREGATION IN THE ENTERPRISE
Social Networks Architecture (SONAR) introduced an approach of aggregating people-to-people relationships across diverse data sources within the organization [5]. SONAR collects from each data source – be it a blogging system, a social network site, or the organizational chart – weighted relationships between employees. Each data source has its own weighting scheme, for example, a blogging system may weight a relationship between two users proportionally to the number of links, comments, and trackback between their blogs. The ultimate result is a rich weighted social graph, which maps relationships between employees in the organization based on various types of data.

An important concept in SONAR is “evidence” – as many of the data sources are public in nature, SONAR can provide an explanation for the weight of the relationship between two individuals. For example, it can be indicated that two individuals are related through 2 joint papers they have co-authored, 3 files they have shared, 5 blog comments they have made to each other, and so on. The aggregated social network information collected from public sources was found to be comparable, and often superior, to relationships extracted from email [4].

PEOPLE RECOMMENDATION
The introduction of the “Do You Know?” widget within the Fringe enterprise SNS site is described in detail in [6] and demonstrates how implicit relationship aggregation can be leveraged to encourage explicit connection. The widget suggests people to connect with in Fringe based on other implicit relationships that indicate familiarity. Its dramatic effect on the number of connections in the Fringe site, and the enthusiastic reaction in the blogosphere pointed at its effectiveness. The number of invitations sent through the widget was almost seven times larger than the number of invalidations sent through the regular profile-based mechanism during the inspected period of time (four months). Moreover, users who have never sent invitations to connect before, neither on Fringe nor on other SNSs, indicated the accurate recommendations prompted them to send invitations for the first time.

The widget also presented explanations for each recommended person through the SONAR “evidence” mechanism – for example, users of the widget could see that a person was recommended due to 3 common friends, 2 patents co-authored together, and a shared manager. Links to the specific common friends, patents, and manager were also provided on demand. This high level of transparency...
was indicated to be highly valuable by the widget's users, increasing trust in the system, helping to understand and justify recommendations and increasing usage in the long term.

Chen et al. [2] also studied people recommendation in an enterprise SNS called Beehive. They compared different algorithms for calculating recommendations, including content-matching, friend-of-a-friend, and a SONAR-based algorithm relying on aggregation, similarly to the one used in the “Do You Know?” Fringe widget. The SONAR algorithm was found to be the most effective, yielding the highest percentage of known people (over 85%), and highest percentage of recommendations rated “good” (over 80%).

Freyne et al. [3] use SONAR to recommend people to new users of the Beehive SNS. As an aggregation system collecting information from sources external to Beehive, SONAR allowed recommendation for brand new users in Beehive whose profile was completely blank. The SONAR-based people recommendations, especially when combined with recommendation for “about you” entry creation, were found to be highly effective in increasing user views and contributions over time. People recommendations were also found to significantly increase user retention rates over time. This work highlights the important capability of relationship aggregation to produce recommendations for new users of a social media application and increase adoption and engagement through these recommendations.

CONTENT RECOMMENDATION

The work in [7] introduced a system for recommending social media items, such as bookmarks, blog entries, and online communities. The recommendations were based solely on the user's social network, as harvested by SONAR. A key distinction was made between familiarity relationships (such as the ones used in the “Do You Know?” widget) and similarity relationships, which imply common social activity, such as co-usage of the same tags and co-commenting on the same blog entries. The familiarity network was found to be more effective in producing interesting recommendations, while the similarity network yielded more unexpected items.

Explanations were provided for each recommended item in the form of showing the people who are related to the recommended items (the "implicit recommenders"). The system could also show the “evidence” for the relationships between the user and the implicit recommenders. These explanations were found to increase interest in items, adding an instant value over the long-term value already shown in [6]. Overall, items recommended based on the familiarity network with explanations yielded a 57% interest ratio.

In another related work, Carmel et al. [1] studied personalized social search based on the user's social network as retrieved through SONAR. Personalization of the search results based on the user's social network was found to be effective, significantly improving over non-personalized results and over topic-based personalization.

CONCLUSION AND FUTURE DIRECTIONS

We summarized several of our previous works that highlighted the value in aggregating social network information and its utilization for highly effective content and people recommendation within enterprise social media. Both people and content recommendations are characterized by a rich knowledge-base in the form of a social graph and a high level of transparency allowing to intuitively explain the recommendations.

Recently, we extended SONAR beyond people-to-people relationships to a new system called Social Networks and Discovery (SaND) [8]. which aggregates all types of relationships among people, documents, and terms. SaND allows combing people-based and content-based analysis, which can enhance recommendation capabilities even further and allow recommendation in context. We also intend to explore recommendation of people who are not familiar to the user, to support extension of one's social capital. Finally, our challenge is to build an aggregation system outside the firewall, dealing with the challenges of scale and identity mapping.

REFERENCES


Pharos: Social Map-Based Recommendation for Content-Centric Social Websites

Wentao Zheng†  Michelle Zhou†  Shiwan Zhao†  Quan Yuan†  Xiatian Zhang†  Changyan Chi†
†IBM Research – China  ‡IBM Research – Almaden

ABSTRACT
Recommendation technologies are widely used in online social websites (e.g., forums and blogs) to help users locate their interests among overwhelming amounts of information. However, it is difficult to make effective recommendations for new users (a.k.a. the cold start problem) due to a lack of user information (e.g., preferences and interests). Furthermore, the complexity of recommendation algorithms may not be easily explained, leaving users with trust issues in recommendation results. To tackle the above two challenges, we are building Pharos, a social map-based recommender system. A social map summarizes users’ content-related social behavior (e.g., reading, writing, and commenting) over time as a set of latent communities. Each community describes the content being discussed and the people involved. Discovering, ranking, and recommending “popular” latent communities on a social map, Pharos enables new users to grasp the dynamics of a social website, alleviating the cold start problem. In addition, the map can also be used as a context for making and explaining recommendations about people and content. We have deployed Pharos internally and the preliminary evaluation shows the usefulness of Pharos.

Author Keywords
Cold start, explanation, recommender systems, social websites, trust

ACM Classification Keywords
H.3.3 Information Search and Retrieval: Information filtering; H.5.3 Information Interfaces and Presentations: Group and Organization Interfaces—Collaborative computing

INTRODUCTION
In recent years, content-centric social websites (e.g., forums, wikis, and blogs) have flourished with an exponential growth of user-generated information. It becomes increasingly more difficult for people to navigate these sites and locate desired information. Thus, researchers have developed recommendation technologies to help people better find desired information at online social websites [4, 5].

However, there are two prevalent challenges when building such a recommender system. First, it is difficult to make effective recommendations if a system knows little about its users or the items to be recommended. This is also known as the cold start problem [10, 11]. In the case of a social website, a recommender system knows little about a user who is new to the website or has few connections to others. As a result, it would be difficult for the system to guess what a user is looking for and make effective recommendations. Second, it is difficult to explain recommendation rationales to end users to make the recommendation more trustworthy. A recommender system usually utilizes complex algorithms or inferences to make recommendations. It is difficult for average users to interpret and comprehend the process and entrust themselves to the recommended results [5, 7].

To address the above two challenges, we are building Pharos, a social map-based recommender system. Here, we use the term social map to refer to a dynamically generated marauder’s map of a content-centric social website. Such a social map summarizes users’ content-related social behavior (e.g., reading, writing, and commenting) over time as a set of “latent communities” (Figure 1-a). Each latent community characterizes the implicit connections among a set of users and the content they generated. It thus consists of two parts, explaining what is being talked about (content summarized in green) and who are involved (people summarized in blue). Based on the generated social map, Pharos then uses a set of criteria (e.g., people’s social status and content popularity) to recommend “hot” communities, “hot” content or people in each community. For example, Figure 1-(b, c) shows the lists of content and people being recommended respectively. As a result, a user especially a new user can use the social map to get a quick glimpse of a content-centric social website and learn about the website’s dynamics (e.g., popular content and people). In this sense, Pharos uses a social map to recommend “hot communities” and relevant “hot items” (i.e., content and people) to all users, alleviating the cold start problem. Furthermore, the social map provides a natural context for users to grasp the recommendation results (e.g., hot communities and hot items), since it helps explain the existence of latent communities and their characteristics (e.g., highly influential content and individuals).

In short, Pharos offers two unique contributions. First, it addresses the cold start problem in part by using a social map to summarize a social website, which in turn helps new users

http://en.wikipedia.org/wiki/Magical_objects_in_Harry_Potter
quickly locate their interests. Second, it uses the social map to help explain recommendation results by means of content, people, and their relationships, helping improve recommendation trustworthiness.

RELATED WORK
Over the years, researchers have used different methods to address the cold start problem. For example, Schein et al. [10] explore several generative probabilistic models to help recommend new items to end users. Park et al. [9] introduce filterbots to make collaborative filtering algorithms more robust when dealing with new items. More recently, Park and Chu [8] take predictive feature-based regression models to tackle three types of cold-start problems: 1) recommendation on existing items for new users; 2) recommendation on new items for existing users; and 3) recommendation on new items for new users. Compared to these works, Pharos provides a social map to new users, which helps them quickly grasp a content-centric web site and locate their interests.

Ensuring recommendation trustworthiness is another popular research topic in building recommender systems. An early piece of work on recommendation explanation is done by Herlocker et al. [7]. They explore the utility of explanations in ACF (automated collaborative filtering) systems, by exposing neighbors and their ratings to end users. Guy et al. [5] study the effect of using social network information to explain recommended items. Vig et al. [13] introduce tag-based explanation, linking items and users via tag relevance and tag preference. Compared to these works, Pharos employs a social map, a summary of users’ content-related social behavior, to help explain recommendation results.

The increasing popularity of social websites leads to various work on social recommender systems. Chen et al. [2] present recommendation algorithms that can effectively expand users’ friend lists in a social network. Guy et al. [4] describe a social recommender that recommends people to join one’s social network. Most existing works rely on explicit social networks to make recommendations. In contrast, Pharos utilizes implicit social relationships dynamically extracted from users’ content-related social behaviors (i.e., latent communities in a social map) to make recommendations.

EXAMPLE APPLICATION
Pharos is designed to help users locate desired information at content-centric social websites, such as forums and blogs. Here, we use the IBM BlogCentral website (our internal blog website) as an example to show the use of Pharos.

Alice, a new employee of IBM Lotus wants to use BlogCentral to learn more about Lotus:

- After login, Alice is presented with a social map of BlogCentral (Figure 1-a). In this case, Pharos chooses to display seven “popular” communities discovered based on user behaviors from August 1st to 15th, 2009. She notices that one community seems talking about “lotus” (highlighted in brown).
- Alice then drills down on the “lotus” community. She gets a list of recommended blog entries (Figure 1-b). Each blog entry is associated with a list of people (e.g., authors and commentors). From the people listed, Alice notices that Bob is a very active blogger. She clicks on Bob to find out more about him. Then she gets a list of recommended bloggers with Bob ranked at the top (Figure 1-c). In addition, the social map highlights Bob’s activities in multiple communities (Figure 2).
- Based on the information displayed on the social map, Alice notices that Bob is also active in another community on the subject of “iphone” (highlighted in yellow in Figure 2). She clicks on that community and then gets a list of recommended blog entries. Again, Alice can now learn more about the content and people involved in the “iphone” community.

In summary, Alice is able to use the social map created by Pharos to understand the key activities of BlogCentral and easily learn the topics and people that she might be interested in.

PHAROS OVERVIEW
Pharos is designed to run on top of existing content-centric social websites. We assume that the target website always
Figure 2. Highlight a user’s activities in multiple communities has a database that contains all the site content and server logs that record user interaction behavior. As shown in Figure 3, Pharos has three key components: social map generator, recommender and visual explainer. Based on users’ content-related social behavior (e.g., reading, writing, and commenting), which are often provided by a website’s database and server logs, Pharos dynamically generates a social map. Based on the generated social map, the recommender uses several algorithms to recommend both people and content to end users. The recommendation is explained visually by means of communities in the context of a social map.

SOCIAL MAP-BASED RECOMMENDATION
Since the creation of a social map is the key of Pharos, here we focus on discussing how Pharos dynamically creates a social map. We also briefly explain how Pharos recommends “hot” content or people in a selected community on a social map.

Social Map Definition
Before describing how Pharos generates a social map and uses it to make recommendations, first we give the formal definition of a social map and its components. A social map is a dynamically generated “marauder’s map” that summarizes users’ content-related social behavior of a content-centric social website. A social map consists of a set of identified latent communities. Each community consists of two parts: the content being discussed and the people involved. Just like communities in a real world, a piece of content or a person may belong to multiple communities at the same time. In addition, a social map evolves over time as people’s behaviors and content dynamically change. More precisely, a social map $M$ consists of $n$ latent communities, $M = \{C_1, \ldots, C_n\}$, where $C_i$ is the $i$-th community. Each latent community $C_i$ has two parts: $C_i = \{D_i, P_i\}$, where $D_i$ represents a set of data/content items; and $P_i$ represents a set of relevant people. Currently, content items in $D_i$ may be considered as user-generated documents (e.g., a blog entry or a forum post).

Social Map Creation
Pharos take three steps to create a social map:

1. community extraction,
2. community ranking, and
3. community labeling.

To extract meaningful latent communities that can form a social map, potentially we can use three approaches: (1) we directly model user-content relationships by using co-clustering methods; (2) starting from people, we group them by their behavioral patterns or demographic characteristics (e.g., geographical information). We then follow people-content relationships to assign associated content to corresponding people groups; (3) starting from the content, we cluster content, and then assign associated people to corresponding content clusters. From our research experience, we decide to adopt the third approach for several reasons. First, due to the extremely sparse (less than 0.1% in our test bed) people-content relationship data, co-clustering methods (approach 1) normally would not work well. Second, a large portion of user behavioral data is contributed by anonymous users, leading to the difficulty of distinguishing different users and modeling user behavioral patterns (approach 2). Third, in a content-centric website, a user is more likely to be interested in content, rather than people. In other words, the third approach of treating the content as the first class citizen and summarizing user social behavior around content makes more sense to end users.

There are many approaches to content clustering. Because of its effectiveness in summarizing text content [3], we decide to leverage LDA (Latent Dirichlet Allocation) [1] to summarize all content into a set of topics. Naturally, topics form content clusters. Based on the probabilistic distributions of each content item over topics, we assign the membership of the content items to the most relevant topic (the probability score being the highest). At last, we assign associated people to the content clusters based on their content-related social behaviors.

Since the derived communities are not ranked, the second step is to rank derived communities based on their importance. The goal of this step is to identify “hot” communities to be presented on a social map. There are a number of metrics to measure the “hotness” of a community. Currently, we measure the “hotness” based on both content and people authorities. The key idea is that content items with more hits and replies are considered more popular, and people who
post more popular content items are considered having more authorities. Similarly, content items written by people with more authorities are more likely to be popular. We build a graph including two types of entities: content items and people. We then link them by various types of user social behaviors (e.g., reading, writing, and commenting). We also consider the attenuation factor of time for each user behavior. Specifically, a more recent behavior is considered more important than the ones in the past. To promote content items with many anonymous visits, which are not modeled by the graph, we adopt the topic-sensitive PageRank [6] by using the amount of hits to bias the topic vector. After calculating authorities for each content item and person, the community’s authority is the averaged summation of authorities of all relevant content items and people.

The final step is community labeling, where each community is to be described informatively so that users can easily understand what a particular community is about. Since each community on a social map describes the relationships between a set of content items and a set of people, the community description must include both types of information (i.e., content and people). Currently, we use LDA-extracted keywords for content summary, and adapt TF/IDF measure [12] to extract representative keywords for people summary, in which people’s profile data is used, such as location and job description. To promote keywords describing “hot” people, the term TF is multiplied by corresponding people authority.

Social Map Display
Based on the rank of communities, Pharos displays the top-$N$ (currently $N \leq 10$) communities along with their descriptions. Our key challenge is to tightly pack multiple communities in one display with adequate community descriptions (content and people labels). Currently, we adopt a bubble chart layout used by ManyEyes\(^2\) to pack top-$N$ communities tightly on a social map. Here each latent community is represented as a “bubble”, and its size is determined by the computed importance. We tend to display more important communities toward the center of the screen. We also use the Wordle\(^1\) layout to tightly pack the related labels in each community.

Content and People Recommendation on Social Map
When a user selects a community on the social map, Pharos recommends two lists of top-ranked content items and people respectively based on their computed authorities. Each content item is associated with multiple people, and vice versa. By default, Pharos shows the list of top-ranked content items. Sometimes, users may want to view the recommended people instead. We are now designing mechanisms that let users interactively specify which list to view.

DEPLOYMENT AND PRELIMINARY EVALUATION
We deployed Pharos internally at IBM since August 2009 to help IBM BlogCentral users navigate blogs and locate their

\(^{1}\)http://www.wordle.net/
\(^{2}\)http://www.manyeyes.com/manyeyes/page/Bubble Chart.html

interests. The back-end system is implemented in Java, and the front-end user interface is implemented in HTML/JavaScript and Flex. To evaluate the usefulness of Pharos, we have conducted a preliminary evaluation.

Our evaluation consisted of two parts. First, we asked eight users to use Pharos at BlogCentral for a short period of time (several weeks). Second, we conducted a survey, asking eleven (11) users including the eight (8) Pharos pilot users about their opinions of BlogCentral and the usefulness of Pharos (if they were Pharos users).

The Survey Results
Our questionnaire contains eight questions, covering three categories: user profiling, viewpoints on current BlogCentral system and recommendations, and assessment of Pharos functions if they have used Pharos before.

From our survey results, 7 out of 11 people selected “read” as their major task in BlogCentral. When asked what they would like to see on the home page of the Blog Central website, 9 out of 11 mentioned that they would like to see an overview of “hot spots”; and 6 out of 11 thought an overview would be very useful. When commenting on the recommendations used by the current BlogCentral website, people’s opinions were inconsistent. Half of them thought they were useful while the other half did not think so. These answers indicated that our motivation in Pharos is valid: it is useful to provide an overview of a social website.

From the eight users who used Pharos for a short period of time, their answers were positive. In particular, they found the social map useful for them to grasp and track the activities at BlogCentral; and the social map also helped explain the recommendation results.

More User Feedback
Our pilot Pharos users also provided us more feedback. Several users commented that there was too much information displayed on a social map, which could be a burden for end users. They suggested that each community only display two or three labels to summarize people and content. On the other hand, a few users complained the labels displayed on a social map were too terse, and sometimes they had to guess what they meant.

CONCLUSIONS
In this paper, we present Pharos, a social map based recommendation, to address two challenges in recommender systems. First, it provides a social overview of a content-centric website along with “hot” content and people, helping new users quickly locate their interests and in part addressing the cold start problem. Second, it uses a social map to explain recommendations by means of relating content with user social activities, increasing recommendation’s trustworthiness. Pharos was deployed internally at IBM, helping bloggers navigate the IBM BlogCentral website. Our preliminary evaluation has showed that the social map helps end users grasp the overview of the website and comprehend recommendations results. Feedback from our pilot users also
points future directions to improve Pharos.

REFERENCES


Conference Navigator 2.0, Community-Based
Recommendation for Academic Conferences

Chirayu Wongchokprasitti  
School of Information Sciences  
University of Pittsburgh,  
Pittsburgh, 15260, USA  
chw20@pitt.edu

Peter Brusilovsky  
School of Information Sciences  
University of Pittsburgh,  
Pittsburgh, 15260, USA  
peterb@pitt.edu

Denis Parra  
School of Information Sciences  
University of Pittsburgh,  
Pittsburgh, 15260, USA  
dap89@pitt.edu

ABSTRACT
As the sheer volume of information grows, information overload challenges users in many ways. Large conferences are one of the venues suffering from this overload. Faced with several parallel sessions and large volumes of papers covering diverse areas of interest, conference participants often struggle to identify the most relevant sessions to attend. This system helps conference participants go through the papers, add most interesting papers to your schedule, and export this schedule to your favorite calendar. To provide further help, the conference planner is a community-based system. The system collects the wisdom of the community in order to help individuals making decisions.

Author Keywords
Information overload, community-based personalization

ACM Classification Keywords
H5.4. Information interfaces and presentation (e.g., HCI): Hypertext/Hypermedia. H3.3. Information storage and retrieval: Information Filtering.

INTRODUCTION
For generations of researchers, paper and pencil were the tools to plan which sessions to attend at an academic conference. To do it right, the researchers were jumping between the conference program and the proceedings to pick up most interesting papers and arrange them into a schedule without context. However, doing it right was a challenging and time-consuming task especially at large multi-stream events. So, many attendees mastered a much easier, yet quite reliable approach: follow the community. It means such heuristics as following researchers who work on similar topics or just going to a room, which seems to be most occupied. In 2006, Farzan and Brusilovsky [3] made an attempt to implement this social navigation approach in an adaptive hypermedia system Conference Navigator (CN) and explore this system at a large conference. This paper presents an update of this project. We introduce Conference Navigator 2.0 (CN2), an attempt to re-implement the original system using ideas and tools provided by Web 2.0 movement. At the time of writing, CN2 was deployed at four conferences: Adaptive Hypermedia (AH) 2008[1], Hypertext (HT) 2009[2], User Modeling, Adaptation, and Personalization (UMAP) 2009[3], and 4th European Conference on Technology Enhanced Learning (EC-TEL) 2009[4]. The use of the system at several real conferences allowed us to collect user feedback, which is also summarized in the paper.

COMMUNITY-BASED PERSONALIZATION
The key idea of Conference Navigator is community-based personalization. By community we mean a relatively small group of people with common interests in respect to the domain of the system. Community-based personalization has been explored in a few search, browsing, and recommender systems [1; 2; 9]. In an academic conference context, a community is a group of people interested in the same, relatively narrow research topic. Depending on its size and focus, a conference can have from a few to many dozens communities. Each attendee may be a part or one or more of these communities. The use of community-level personalization places CN and CN2 between traditional social navigation systems (which offer the same guidance to all system users) and recommender systems (which adapt to individual users). In other words, CN and CN2 provide personalization on a group level [5].
COMMUNITY-BASED CONFERENCE NAVIGATOR

CN2 was an attempt to redesign a community-based conference navigator as a Web 2.0. It is designed as a conference schedule planner service, which allow conference participants browse the conference program and plan their schedule. The system supports the users with social guidance, which is provided through bookmarks of communities and individual attendees. CN2 is built using the Google Web Toolkit (GWT) package. The GWT provides a new paradigm of web usage perspective. It provides the easier way to implement a web-based application to have the same look-and-feel as usual desktop applications. GWT gives CN2 its multi-tab look-and-feel.

Figure 1 The Hypertext 2009 Conference Summary

Conference Summary

The conference summary tab plays the role of a home page in CN2. It consists of 5 small gadgets: “Top Ten Annotation Papers”, “Top Ten Viewed Papers”, “Tag Cloud”, “Active Users”, and “Top Ten Active Communities”. The top ten annotated papers gadget counts a number of bookmarks users make and shows top ten of the list. The top ten viewed papers gadget does the same thing but counting views users visit papers. Tag cloud gadget presents the cloud of tags users make bookmarks on papers and add tags to their bookmarks. Tag cloud also provides the metadata of the whole conference. Active users gadget provides a cloud of users who make schedule on the system. The last one, top ten active communities gadget, represents top ten of communities counting on bookmarks users contribute to that community.

Program Browser

The program browser presents the entire program of the conference divided into 2 parts. The upper part shows the abstract of the conference and the lower presents the program detail, which is separated by the day of each session. Each day session application panel provides the table of all sessions ordered by time. Also, on the upper right corner, the system provides a link to export a conference program to an iCal file.

Figure 2 Program Browser

When users browse on each session by clicking on the session record. Like program detail tab panel, the session tab panel consists of 2 parts: the upper part shows the abstract or detail of that session and the lower part shows the table of papers ordered by the time presenting. To view the detail of each paper, click on a paper record row to show Paper Panel in another tab.

Figure 3 Paper Panel

Paper Panel

The Paper Panel has 2 columns. The left one consists of 2 parts: the upper part shows the paper detail and the lower part provides the list of notes of other users. The right column provides the bookmark, annotate, and contribute feature. Note for the bookmark, annotate, and contribute feature, the system initiated an amount of communities the experts thought they were relevant to the conference. User can add more communities in case there was not in the list. This feature would let users bookmark a paper, add tags, leave a note, and also contribute a paper to their
communities. For HT’09 and EC-TEL’09, the system integrated a new feature that let users export their papers to Bibsonomy social bookmarking and publication sharing system [4].

Figure 4 Bookmark, Annotate, and Contribute Feature

Schedule Panel
There are two types of Schedule Panel: User Schedule Panel and Community Schedule Panel. Schedule Panel is to provide schedule in detail in the left column boxes and summary in the right.

User Schedule Panel
Papers in schedule are sorted by date and time. Paper box shows user’s note, tags, and contributed communities. In “My schedule” also has an extra “Delete” button on lower right of paper box to let you remove your marking. On the right column, the user schedule provides a link to export the user schedule to an iCal-format calendar file. Also, it provides groups of social wisdom cloud and grid: tag cloud, co-bookmarking users, and communities.

Community Schedule Panel
The functionality of the community schedule panel is the same as user schedule, but rather represents community’s interest. The community schedule provides the users the ability to see community’s schedule.

Figure 5 User Schedule Panel

PRELIMINARY EVALUATION AND RESULT
To evaluate the system, we deployed it at four large academic conferences and solicited user feedback through participants’ questionnaire.

User Study
We collected users’ activities on the system. From system log, we can have the overview statistics result as below.

Figure 6 Community Schedule Panel

Log analysis
To judge how extensively the system was used in our trials, we processed system’s logs. The logs showed that system users explored a considerable fraction of all papers through the system: 39.47% (45 out of 114), 50.51% (49 out of 97), 69.92% (93 out of 133), and 64.43% (96 out of 149) for AH’08, HT’09, UMAP’09, and EC-TEL’09, respectively. The data hints that the conference complexity (from planning point of view) does impact system’s usage. The conference with a large number of papers and many parallel sessions (ECTEL’09) provides the strongest motivation to
use CN2 for schedule planning. Or discussions with attendees at several events confirm this observation.

Log analysis also provide some evidences in favor of system’s usefulness in discovering good papers. When users viewed papers, they tended to bookmark them. From the data, users bookmarked papers 71.11% (32 out of 45), 95.92% (47 out of 49), 86.02% (80 out of 93), and 79.17% (76 out of 96) for AH’09, HT’09, UMAP’09, and EC-TEL’09, respectively. For HT’09, even though, the size of conference was not that big but there were many parallel sessions, which made HT’09 more complex. As a result, users viewed not much but they bookmarked more percentage than UMAP’09, which there was only one track.

From bookmarked papers, users were more likely to share the wisdom to communities. They contributed bookmarks to their communities they thought bookmarks relevant to 78.13% (25 out of 32), 97.87% (46 out of 47), 70% (56 out of 80), and 72.37% (55 out of 76) for AH’09, HT’09, UMAP’09, and EC-TEL’09, respectively.

From bookmarking users, users were more likely to share the wisdom to communities. They contributed bookmarks to their communities they thought bookmarks relevant to 78.13% (25 out of 32), 97.87% (46 out of 47), 70% (56 out of 80), and 72.37% (55 out of 76) for AH’09, HT’09, UMAP’09, and EC-TEL’09, respectively.

From bookmarking users, users were more likely to share the wisdom to communities. They contributed bookmarks to their communities they thought bookmarks relevant to 78.13% (25 out of 32), 97.87% (46 out of 47), 70% (56 out of 80), and 72.37% (55 out of 76) for AH’09, HT’09, UMAP’09, and EC-TEL’09, respectively.

From bookmarking users, users were more likely to share the wisdom to communities. They contributed bookmarks to their communities they thought bookmarks relevant to 78.13% (25 out of 32), 97.87% (46 out of 47), 70% (56 out of 80), and 72.37% (55 out of 76) for AH’09, HT’09, UMAP’09, and EC-TEL’09, respectively.

For a point of view of system developers, good systems should encourage and convince users to participate in the activities of the systems. In the other word, a number of free riders would be considered, too. From the data, users just only viewed papers were 34.78% (8 out of 23), 21.74% (5 out of 23), 37.93% (11 out of 29), and 3.70% (1 out of 27) for AH’09, HT’09, UMAP’09, and EC-TEL’09, respectively. From these percentages, the data confirmed that the complexity matters. Users from EC-TEL’09 were more intended not just to view papers but rather bookmarked them, too. Comparing to 3 other conferences, which were less complex of both of a number of papers and a number of parallel sessions, there were more number of free riders. We think more study about free riders need to be addressed.

**Questionnaire**

To evaluate our community-based conference navigator system, we ran a user study at the 4 conferences as mentioned before. We asked participants to plan the conference using the system. The participants were also asked to respond to a questionnaire, which was designed to evaluate the community-based support features of the system. We collected the questionnaire after the conference finished. The questionnaire included 12 short questions for AH’08, 15 for HT’09, 13 for UMAP’09, and 13 for EC-TEL’09. A number of questions vary from the features we implemented more for specific conference.

**Social Annotations Usefulness and Attractiveness**

A set of questions was asked about the usefulness and attractiveness of social annotations. 80% of them found the social annotations easily noticeable. 75% agreed that social tags were useful. 55% were neutral that social comments were useful. Generally, they found the social annotations quite useful in planning the schedule.

**Navigational Tools/Gadgets Usefulness**

The next set of questions asked about usefulness of each navigational tools/gadgets in the conference summary and personal schedule. In the conference summary, users agreed that the presence of top ten of annotated and visited papers were useful. Also, they quite strongly agree that tag cloud was more useful. Unsurprisingly, the presence of active users and top ten of active communities were neutral to them. Users were more interested in knowing which paper was more important to other users. In personal schedule, 75% of users agreed that tag cloud was useful. 70% of them agreed that co-bookmarking users gadget was useful. They found neutral to the presence of related communities.

**Calendar Exporting Features**

Another set of questions was about the exporting feature and ability to hide some information. We provided the exporting a paper to Bibsonomy at HT’09 and EC-TEL’09. Users from both conferences were quite neutral for this feature. Also, they found neutral to the exporting schedule to iCal feature. Not everyone was using iCal. Consistently, they found neutral that hiding tags to other users was useful. But they quite a bit agreed that hiding comments to others was useful, which was not surprised to us.

**Comments/Suggestions to Conference Navigator**

The last question is open question asking comments or suggestions about the system. Some users complained about the performance of the system, for example, one from
UMAP’09 said, “Performance could be improved - loading the page sometimes was a drag” and one from EC-TEL’09 said, “The software is very slow and thus using it is not easy”. Many users suggested us to provide a mobile version of the system, such as ones from HT’09: “I was also using a mobile device that didn’t support the AJAX portal. For the next version, it would be useful to have a version compatible with mobile devices”, and “Provide a version for download on the mobile phone, since I didn’t use much of it while at the conference”, and one from EC-TEL’09 said, “The conference planner needs to be available in advance of the conference, as when I go, I will have made up at least 50% of my schedule already. I would prefer an iPhone app to a website”. Some users said the system was useful but amount of users was small, “I think the system may have been useful, but the small user group making use of the system limited this usefulness ...”.

CONCLUSION & FUTURE WORK
The current work presents our design for a community-based conference navigator system that collects the wisdom of the community in order to guide individuals making decisions about attendance at papers presented at a large conference. We have presented the design and an evaluation of the system. We plan to implement with other approaches such as content-based filtering [6], collaborative filtering recommendation [8], tag-based recommendation [7] and so on. We are working on improving the performance of the system since users complained about the slow response. Another thing is to create a mobile version, because users do not have much time to use laptop. As a result, integrating a system with a mobile application is a complement.

REFERENCES