Complex Adaptive Filtering User Profile Using Graphical Models

Yi Zhang [*]
School of Engineering
University of California Santa Cruz
1156 High Street
Santa Cruz, CA, USA
yiz@soe.ucsc.edu
Tel: 1-831-459-4549
Fax: 1-831-459-4826

August 5, 2008

Abstract

Abstract

This article explores how to develop complex data driven user models that go beyond the bag of words model and topical relevance. We propose to learn from rich user specific information and to satisfy complex user criteria under the graphical modelling framework. We carried out a user study with a web based personal news filtering system, and collected extensive user information, including explicit user feedback, implicit user feedback and some contextual information. Experimental results on the data set collected demonstrate that the graphical modelling approach helps us to better understand the complex domain. The results also show that the complex data driven user modelling approach can improve the adaptive information filtering performance. We also discuss some practical issues while learning complex user models, including how to handle data noise and the missing data problem.

Key Words

Information filtering, adaptive user modelling, graphical models

1 Introduction

An adaptive personal information filtering system is an autonomous agent that delivers information to the user in a dynamic environment over a period of time. The study of user profile learning is central to the filtering research. A common approach is to learn the user profile as a classifier, by adapting existing text classification/retrieval algorithms to classify incoming documents

[*] Part of the work was carried out when the author was at Carnegie Mellon University.

as either relevant or non relevant. New documents that are similar to relevant documents the user has seen before are usually delivered to the user. However, this approach is a very simple and limited view of user modelling. It can not address many practical issues such as the complex user criteria (for example, novelty (Harman, 2003)). Also, it is not clear how to fully take advantages of the contextual information, implicit and explicit user feedback that can be collected by a filtering system.

This study explores how to go beyond the bag of words model and topical relevance based filtering. To use rich user specific information and satisfy complex user criteria, our approach is to represent the filtering system's belief about each user, which is learned from multiple forms of evidence, as a probabilistic graphical user model. Our hypothesis is that the user models can be used in two directions. First, the models can provide guidance for the system analysist/designer, for example, helping the system designer to decide whether to collect or use certain user information. Second, the models can be directly used in the choice of a system action, for example, helping the system to decide whether to deliver a document to the user.

To test the hypothesis and explore the graphical modelling approach in both directions, we first carried out a user study to collect a new evaluation data set that contains thousands of extensive implicit user feedback (such as a user's mouse usage, keyboard usage, document length), explicit user feedback about the news (such as novelty, relevancy, readability, authoritativeness, and whether a user likes a document or not) along with other forms of evidence (such as news source information). We performed several experiments with the data collected. The experiments were designed to explore the potential of graphical models in the above two directions and answer the following two specific questions: 1) Can the graphical modelling approach help us better understand the domain? 2) Can the graphical modelling approach help us improve the performance of an adaptive information filtering system?

The following sections report our efforts to answer the above questions. We begin with a review of some related work in Section 2. Section 3 introduces the graphical modelling approach and how it could be used for adaptive filtering profile learning. Section 4 describes our efforts towards collecting the new adaptive filtering data set and evaluating graphical models for the task of developing complex user models for filtering. Section 5 concludes our findings and discusses future work.

2 Background and Related Work

The goal of an information retrieval system is to find relevant information. The definition of relevance is the fundamental problem when designing and evaluating an IR system. Most of the standard evaluations are based on a narrow definition of "topical relevance" or aboutness (Voorhees & Buckland, 2002). Recently, researchers have studied criteria beyond topical relevance. (Carbonell & Goldstein, 1998) studied combining query-relevance with information-novelty for retrieval and summarization and proposed Maximal Marginal Relevance (MMR) criterion to reduce redundancy. (Varian, 1999) considered the incremental value of a piece of information and argued that the standard way that presents documents "in order of estimated relevance" is not appropriate. (Zhang *et al.*, 2002) proposed a two stage filtering system to filter out relevant but redundant documents. (Zhai *et al.*, 2003) went beyond independent relevance to model dependent relevance to retrieve documents that cover as many different subtopics as possible. (Wang, 1994) asked users to read aloud and think aloud while doing hard copy documents selection. After analyzing audio recording of the whole process, they proposed a relevance model based multiple criteria, such as personal knowledge, topicality, quality, novelty, recency, authority and author. (Schamber & Bateman, 1996) identified criteria underlying users' relevance judgments and explored how users employed the criteria in making evaluations by asking users to interpret and manually sort the criteria independent of documents. In previous work the word "relevant" was used ambiguously, either as a narrow definition of "related to the matter at hand" (aboutness) or a broader definition of "having the ability to satisfy the needs of the user". When the second definition is used, such as in (Schamber & Bateman, 1996), researchers were usually studying what this paper refers to as *user_like* or likability. In this paper, we use the first definition of relevance and use *user_like*, or likability, for the second definition. Despite the vocabulary difference, the work in this paper is motivated by these early works focused on "likability". The work reported in this paper goes beyond relevance by 1) modelling the likability and other criteria as hidden variables; 2) quantifying the importance of various criteria based on probabilistic reasoning; and 3) combining these criteria with implicit and explicit user feedback in a single graphical model.

Now, IR research is moving into a context and user dependent scenario. For examples, SearchPad treated the previous information requests from the user as the context of the current query to improve retrieval results (Bharat, 2000), and (Shen *et al.*, 2005) treated the preceding queries and clicked document summaries as the context of the current query. Not necessarily in the context of personalization, there is much prior research on using implicit feedback in the information retrieval community and user modelling community. (Kelly & Teevan, 2003) provided a review and classification of works in these areas according to the behavior category and minimum scope. There is also much related work on using implicit feedback to improve web retrieval performance (White *et al.*, 2006) (Anderson & Horvitz, 2002) (Sugiyama *et al.*, 2004). These prior efforts suggested many possible behaviors (view, listen, scroll, find, query, print, copy, paste, quote, mark up, type and edit) on different scope (segment, object and class) for system designers to use as implicit feedback. However, much of the earlier work on personalization are incremental improvement in existing similarity based models, which will not be enough to address many practical issues such as confidence, privacy, authority, novelty, recency, long term and short term retrieval personalization (Callan *et al.*, 2003). How to use the rich personal and contextual information in a principled way to better satisfy the user's information needs in a complex environment becomes an important problem.

There is much prior research on news customization (Lang, 1995) (Ardissono *et al.*, 2001) (Domingue & Scott, 1998) (Henzinger *et al.*, 2003) (Carreira *et al.*, 2004) (Lai *et al.*, 2003) (Merialdo *et al.*, 1999). (Billsus & Pazzani, 1999) built a personal news agent that used time-coded feedback from the user to learn a user profile. However their way of using time as feedback is rather heuristic. (Morita & Shinoda, 1994) investigated implicit feedback for filtering news group articles. They treat reading articles for more than 20 seconds as positive feedback, and they found this can produce better recall and precision than user's explicit rating.

Different graphical models, such as Bayesian networks, dependency networks, inference networks, and causal models, have been used to model computer software users (Horvitz *et al.*, 1998), car drivers (Pynadath & Wellman, 1995), students (Conati *et al.*, 1997) and other social phenomena (McKim

& Turner, 1997). Choosing the graphical modelling approach as a unified framework to combine multiple forms of evidence is motivated by the prior research. Recently, there has been some independent work using a different graphical modelling approach (dependency networks) to discover the relationships between implicit measures and explicit satisfaction, and using decision tree for prediction (Fox *et al.*, 2005). They were focusing on predicting user satisfaction with web search results based on implicit measures gathered while users were conducting their searches and viewing results. Their findings justify the graphical modelling approach's effectiveness in a closely related task. Our work differs from the previous work in that: we are focusing on the *adaptive filtering* task instead of web search; we carried out a detailed user study with human subjects in a news recommendation setting; we develop graphical models with very different structure and functional form, which will be discussed in detail later; we want to satisfy complex user criteria; we consider a very different set of explicit feedback, implicit feedback and contextual information for user modelling; and the findings and conclusions we reached are very different.

3 Graphical Models for Adaptive Complex User Modeling

The basic methodology behind the graphical modelling approach is to represent the system's belief about a user as a graph that summarizes the conditional dependence relationships between user history and context. Each node in the graph represents a random variable, and each arc represents a conditional dependence between the variables. Variables that do not share an arc are conditionally independent given other variables. The graph can be either directed, such as Bayesian Networks, or undirected, such as Markov Random Fields. We will focus on the directed graph in this paper. A graphical model includes the definition of the graph structure and a set of local conditional probability functions or potential functions. Structure learning and probabilistic inference are the two key techniques while using graphical models.

Automatically learning the structure of the graph from the data can be achieved through two major approaches. The first approach assigns a score, such as the likelihood of the training data given the structure, to each candidate graph. Usually a structure with the best score is selected. The second approach finds some constraints and keeps the causal graph(s) consistent with these constraints are as valid. Besides the constraints automatically generated from the data, a person can also specify prior constraints based on domain knowledge.

In a graph, if there is an arc from node X to node Y if and only if X is a direct cause of Y, then we call the graph a *causal graph*. One of the major goals and advantages of structure learning is the ability to automatically learn the *causal graph* that encodes the causal relationships between variables. This will help us to understand the problem domain and answer questions, such as whether liking a document causes increased reading time, or whether the authority of a page is important to the user. Some structure learning algorithms try to achieve the goal of causal discovery directly. These algorithms are new and subject to criticisms. However, because of the potential of these algorithms, their success in some domains (McKim & Turner, 1997) (Pearl, 2000) (Spirtes *et al.*, 2000), and the lack of causality based analysis in the information retrieval community, we decided to introduce this important technique to the IR community and apply it to the task of filtering in this paper.

As an example, let's look at a simple constraint based causal learning algorithm that will be used later in Section 4.3: *PC algorithm* (Spirtes *et al.*, 2000). To learn the causal structure, the PC algorithm begins with a complete graph, finds zero order conditional independence relations from the data, then removes edges that contradict the relations. The algorithm continues with first order conditional independence relations, and so on. Finally, the algorithm finds some head to head links, and orients the links without producing cycles. The algorithm finds a set of models that can not be rejected on the basis of the data. This algorithm is computationally efficient with polynomial time complexity. However, it assumes no hidden common causes, and the causal relationships are acyclic. These assumptions, especially the assumption of no hidden variables, may not hold in the real scenario.

Some algorithms make fewer assumptions. For example, the Fast Causal Inference algorithm (FCI) handles unmeasured hidden variables (Spirtes *et al.*, 2000). (Pearl, 2000) and (Spirtes *et al.*, 2000) provided extensive detailed descriptions about learning structures with causal meanings, including hidden variables, cycles, and undirected graphs. More details of causal structure discovery are beyond the scope of this dissertation, and we refer the reader to these books for more information on this topic.

Probabilistic inference is one of the most important step in the graphical modelling approach. Probabilistic inference means computing the conditional probabilities $P(x_F|x_E)$, where E are observable variables, and F are unobservable variables we need to estimate. There are several different algorithms for probabilistic inference, such as exact algorithms (Pearl, 1988), sampling based algorithms (Tanner, 1996) (Thomas *et al.*, 1992) (Mackay, 1998), variational algorithms (Jaakkola & Jordan, 2000) (Jordan *et al.*, 1999), most likely configuration, parametric approximations (Minka, 2001) (Yedidia *et al.*, 2000), and heuristic methods (Heckerman *et al.*, 2000). Different algorithms have different trade-offs between computational speed, implementation complexity, generality and accuracy.

Researchers have identified three major advantages for the graphical modelling approach. First, it provides inference tools to naturally handle situations of missing data because of the conditional dependencies encoded in the graph structure. Second, it can learn causal relationships in the domain, thus helping us to understand the problem and to predict the consequences of intervention. Third, it can easily combine prior knowledge (such as partial information about the causal relationship) with data. This approach has been applied to model computer software users (Horvitz *et al.*, 1998), car drivers (Pynadath & Wellman, 1995), and students (Conati *et al.*, 1997).

Because of these advantages, we hypothesize that the graphical modelling approach is a useful tool for filtering tasks, where we have complex user criteria, implicit and explicit user feedback, and contextual information. The filtering system's belief about the user can be represented as a graphical model. The belief may be used in two ways, either guiding the system designer's future actions (such as deciding whether to collect certain evidence), or directly guiding the choice of a system action (such as deciding whether to deliver a document to the user).

4 Experiments

To test the above hypothesis and explore the graphical modelling approach in both directions, we carried out several experiments. The experiments were designed to answer the following two specific questions:

- Can the graphical modelling approach help us to better understand the domain? For examples, can the algorithm tell us what the relationships are between user actions and relevance of a document, how authority relates to the user preference for the page, whether the usage of a specific keyboard key is informative, how users differ from each other, and so on. This information may guide us in designing a better filtering system.
- Can the graphical modelling approach help us to improve the performance of an adaptive information filtering system? For example, when a document arrives, can we better predict a user's preference for the document? This prediction will be used directly in deciding whether to deliver the document to the user.

To answer the above questions, we exploit the three advantages of graphical models in the experiments. More specifically, to see whether the proposed solution can help us to understand the domain better, we use the causal graph structure learning algorithms (advantage 2), together with some prior knowledge of the domain (advantage 3), to derive the causal relationships between different user feedback, actions and user context. To see whether the proposed solution can help us to improve an existing filtering system, especially in the situation of missing data, we use statistical inference tools to predict how much a user likes a document under different evidence missing conditions (advantage 1). Different graphical models are developed and evaluated for different purposes, either to understand the domain or to improve the prediction accuracy. Linear regression is also tried as an alternative approach to combine multiple forms of evidence.

4.1 Evaluation Data

No existing filtering database contains the level of detail that we needed for our study, so we developed a web based news story filtering system to collect an evaluation data set (Figure 1). The system included a crawler that constantly gathers information from 8000 candidate RSS news feeds (Pilgrim, 2002). The Lemur indexer indexed the crawled document stream incrementally (Croft *et al.*, 2004), and an adaptive filtering system constantly recommended documents to the users using a modified logistic regression algorithm (Zhang, 2004). Users read and evaluated the delivered documents. An example of the web interface after user login is shown in Figure 2.

21 paid subjects from 19 different programs at Carnegie Mellon University participated in the study for 4 weeks. The subjects are otherwise not affiliated with our research. We expected to collect enough data for evaluation over this period of time. The subjects were required to read the news for about 1 hour per day and provide explicit feedback for each page they visited.[1] 28 users tried this system. However, only 21 users were official paid subjects, among which one worked only for 2 weeks and 20 worked for about 4 weeks.

[1] In the last week of the study, some subjects read 2 hours per day. They are encouraged but not required to do so.

Figure 1: The user study system architecture. The structured information, such as user feedback and crawler statistics, are kept in the database. The content of each web page crawled is saved in the news repository.

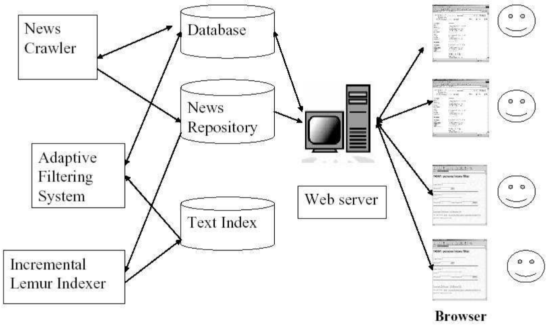
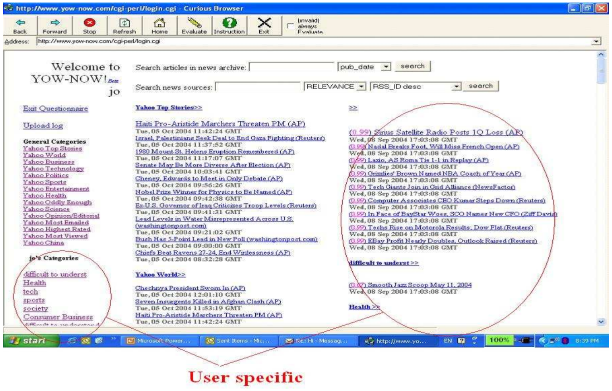


Figure 2: Web interface after a user logged in.



Address: http://www.yow-now.com/cgi-perl/login.cgi

Welcome to YOW-NOW! Beta
jo

Search articles in news archive: pub_date search

Search news sources: RELEVANCE RSS_ID desc search

Exit Questionnaire

Upload log

General Categories
Yahoo Top Stories
Yahoo World
Yahoo Business
Yahoo Technology
Yahoo Politics
Yahoo Sports
Yahoo Entertainment
Yahoo Health
Yahoo Oddly Enough
Yahoo Science
Yahoo Opinion/Editorial
Yahoo Most Emailed
Yahoo Highest Rated
Yahoo Most Viewed
Yahoo China

jo's Categories
difficult to underst
Health
tech
sports
society
Consumer Business
difficult to understand

Yahoo Top Stories>>

Haiti Pro-Aristide Marchers Threaten PM (AP)
Tue, 05 Oct 2004 11:42:24 GMT
Israel, Palestinians Seek Deal to End Gaza Fighting (Reuters)
Tue, 05 Oct 2004 11:37:52 GMT
1980 Mount St. Helens Eruption Remembered (AP)
Tue, 05 Oct 2004 11:17:07 GMT
Senate May Be More Diverse After Election (AP)
Tue, 05 Oct 2004 10:03:41 GMT
Cheney, Edwards to Meet in Only Debate (AP)
Tue, 05 Oct 2004 09:56:26 GMT
Nobel Prize Winner for Physics to Be Named (AP)
Tue, 05 Oct 2004 09:42:38 GMT
Ex-U.S. Governor of Iraq Criticizes Troop Levels (Reuters)
Tue, 05 Oct 2004 09:41:31 GMT
Lead Levels in Water Misrepresented Across U.S. (washingtonpost.com)
Tue, 05 Oct 2004 09:21:02 GMT
Bush Has 5-Point Lead in New Poll (washingtonpost.com)
Tue, 05 Oct 2004 09:00:00 GMT
Chiefs Beat Ravens 27-24, End Winlessness (AP)
Tue, 05 Oct 2004 08:32:28 GMT

Yahoo World>>

Chechnya President Sworn In (AP)
Tue, 05 Oct 2004 12:01:10 GMT
Seven Insurgents Killed in Afghan Clash (AP)
Tue, 05 Oct 2004 11:53:19 GMT
Haiti Pro-Aristide Marchers Threaten PM (AP)
Tue, 05 Oct 2004 11:42:24 GMT

>>

(0.99) Sirius Satellite Radio Posts 1Q Loss (AP)
Wed, 08 Sep 2004 17:03:08 GMT
(0.99) Nadal Breaks Foot, Will Miss French Open (AP)
Wed, 08 Sep 2004 17:03:08 GMT
(0.99) Lazio, AS Roma Tie 1-1 in Replay (AP)
Wed, 08 Sep 2004 17:03:08 GMT
(0.99) Grizzlies' Brown Named NBA Coach of Year (AP)
Wed, 08 Sep 2004 17:03:08 GMT
(0.99) Tech Giants Join in Grid Alliance (NewsFactor)
Wed, 08 Sep 2004 17:03:08 GMT
(0.99) Computer Associates CEO Kumar Steps Down (Reuters)
Wed, 08 Sep 2004 17:03:08 GMT
(0.99) In Face of BayStar Woes, SCO Names New CFO (Ziff Davis)
Wed, 08 Sep 2004 17:03:08 GMT
(0.99) Techs Rise on Motorola Results, Dow Flat (Reuters)
Wed, 08 Sep 2004 17:03:08 GMT
(0.99) EBay Profit Nearly Doubles, Outlook Raised (Reuters)
Wed, 08 Sep 2004 17:03:08 GMT

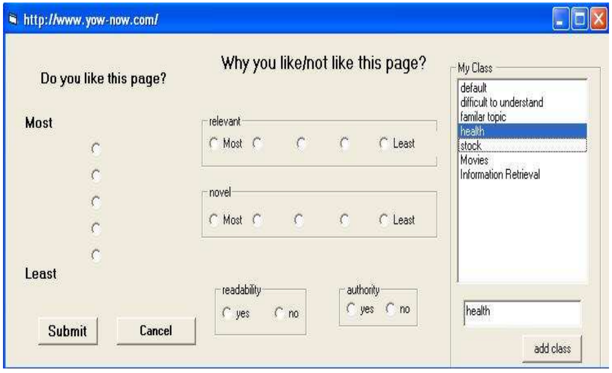
difficult to underst >>

(0.67) Smooth Jazz Scoop May 11, 2004
Wed, 08 Sep 2004 17:03:08 GMT

Health >>

User specific

Figure 3: Evaluation user interface. The interface for user to give their explicit feedback of the current news story.



Do you like this page?

Most
○
○
○
○
○
Least

Why you like/not like this page?

relevant
○ Most ○ ○ ○ ○ Least

novel
○ Most ○ ○ ○ ○ Least

readability
○ yes ○ no

authority
○ yes ○ no

My Class
default
difficult to understand
familar topic
health
stock
Movies
Information Retrieval

health

We have collected 7881 feedback entries from all 28 users, among which 7839 were from the 21 official participants. Each entry contained several different forms of evidence for each news story a user clicked.[2] Our intention to collect the evidence was not to be exhaustive, but representative. The evidence can be roughly classified into the following five categories listed in Tables 1 to 5.[3]

Explicit User Feedback After finishing reading a news story, a user clicked a button on the toolbar of the browser to bring up an evaluation interface shown in Figure 3. Through this interface, the user provided the explicit feedback to tell the hidden properties about current story, including the topics the news belongs to (*classes*), how the user liked this news (*user_like*), how relevant the news was related to the class(es) (*relevant*), how novel the news is (*novel*), whether the news matched the readability level of the user (*readable*), and whether the news was authoritative (*authoritative*). *user_like*, *relevant* and *novel* were recorded as integers ranging from 1 (least) to 5 (most). *readable* and *authoritative* were recorded as 0 or 1. A user has the option to provide partial instead of all explicit feedback. A user could create new classes, and choose multiple classes for one documents.

User Actions The special browser is developed based on (Claypool *et al.*, 2001). It recorded some user actions, such as mouse activities, scroll bar activities, and keyboard activities (Table 2). TimeOnPage is the number of seconds the user spent on a page, and EventOnScroll is the number of clicks on the scroll bars. When the mouse is out of the browser window or when the browser window is not focused, the browser does not capture any activities. More details about the actions are in (Le & Waseda, visited Oct. 2006).

Topic Information Each participant filled out an exit questionnaire and answered several topic/class[4] specific questions for each of his/her 10 most popular topics alone with other topics that have more than 20 evaluated documents each (Table 3). The questions included how familiar the user was with the topic before the study (*topic_familiar_before*), how the user liked this topic (*topic_like*), and how confident the user was with respect to the answers he/she provided (*topic_confidence*). We included this information as evidence, because they may be collected when a topic is created and used by a filtering system. Whether collecting the answer in the exit questionnaire affected the answers needs further investigation.

News Source Information For each news source (RSS feed), we collected the number of web pages that linked to it (*RSS_link*), the number of pages that linked to the server that provided it (*host_link*), and the speed of the server that hosted it.

Content Based Evidence Three pieces of evidence were collected to represent the content of each document: the relevance score, the readability score and the number of words in the document (*doc_len*) (Table 5). To estimate the relevance score of a document, the system processed all the documents a user put into a class, ordered by the feedback time. It then adaptively learned a topic specific relevance model using the

[2] Each entry is for a <document, user class, time> tuple.

[3] The forms of evidence are listed in the first column and we will get to the other columns later in Section 4.2.

[4] "topic" and "class" are used interchangeably in the paper.

relevance feedback the user provided. The relevance score of a document was estimated using a modified logistic regression model learned from all feedback before it (Zhang, 2004). To estimate the readability score of document, the system processed all the documents in all users' classes, ordered by the feedback time. It then adaptively learned a user independent readability model using a logistic regression algorithm.

4.2 Preliminary Data Analysis

The means and variances of all variables are in Tables 1 to 5. These basic descriptive statistics are very diverse. The values of some evidence may be missing; only the user actions and news source information were always collected. Out of the 7,991 entries, only 4,522 (57%) entries contain no missing values. The missing rate of each form of evidence is also reported in the tables. There are several reasons for the missingness. For example, the explicit feedback is missing because users didn't always follow instructions, the relevance score is missing for the first story in a class, and the *topic_familiar_before* values for many topics are missing because we only collected the topic specific answers for larger topics. We expect missing data to be common in operational environments.

The correlation coefficient between each piece of evidence and the explicit feedback *user_like* is also listed (*corr*). The high correlation coefficients between *user_like* and other forms of explicit feedback are not very interesting because we can only get explicit feedback after a user reads the document. The correlation coefficient between the relevance score and *user_like* is 0.37, the highest among all forms of evidence that the system can get before delivering a document. This is not surprising since relevance is a major factor that influence *user_like* judgements, while relevance score is the system's estimation of how relevant a document is.

The correlation coefficients between *user_like* and the topic information (Table 3) are relatively high. This suggests asking a user how familiar he is with a topic created (*topic_familiar_before*) or how much he likes a topic (*topic_like*) in a real filtering system would be helpful. Collecting this data requires little user effort, since a user only needs to provide information on the class level instead of document level. Section 4.4 will show how to use the information with other forms of evidence in a filtering system. The correlation coefficients between the news source information and *user_like* are weaker (Table 4). The correlation coefficient between *user_like* and each user action (Table 2) is even lower (Table 1). Some actions, such as *TimeOnPage*, are more correlated with *user_like* than other refined actions, such as *NumOfPageDown*. This finding agrees with (Claypool *et al.*, 2001) and (Morita & Shinoda, 1994).

4.3 Understanding the Domain Using Causal Structure Learning

In order to better understand the domain, we need to go beyond correlation and investigate the potential causal relationships between different variables. To accomplish this, PC algorithm (Section 3) is used to automatically learn a causal graph from the data collected. Before running the PC algorithm, we specify the prior knowledge developed by the author as temporal-tier constraints of variables before automatic structure learning:

Level 1, 3 nodes: Topic information ($topic\_info =$<*familiar_topic_before*>),

Table 1: Basic descriptive statistics about explicit feedbacks.

Variable	Mean	variance	corr	miss
user_like	3.5	1.2	1	0.05
relevant	3.5	1.3	0.73	0.005
novel	3.6	1.33	0.70	0.008
authoritative	0.88	0.32	0.50	0.065
readable	0.90	0.30	0.54	0.012

Table 2: Basic descriptive statistics about user actions. The unit for time is second.

Variable	Mean	variance	corr
TimeOnPage	7.2×10^4	1.3×10^5	0.14
EventOnScroll	1	3.6	0.1
ClickOnWindow	0.93	2.5	0.05
TimeOnMouse	2×10^3	5.8×10^3	0.02
MSecForDownArrow	211	882	0.08
NumOfDownArrow	1.1	4.7	0.09
MSecForUpArrow	29	240	0.03
NumOfUpArrow	0.10	0.8	0.04
NumOfPageUp	0.12	0.9	$\simeq 0$
NumOfPageDown	0.14	1	$\simeq 0$
MSecForPageUp	22	202	$\simeq 0$
MSecForPageDown	28	251	$\simeq 0$

Table 3: Basic descriptive statistics about topics. Each variable ranges from 1 to 7.

variable	Mean	variance	corr	miss
topic_familiar_before	3.6	1.9	0.30	0.27
topic_like	4.9	2.0	0.30	0.27
topic_confidence	4.7	2.0	0.34	0.27

Table 4: Basic descriptive statistics about news sources.

variable	Mean	variance	corr
RSS_link	90.35	4.89	0.14
host_link	4.41×10^4	7.5×10^7	0.08
RSS_SPEED	3.92×10^5	3.7×10^9	-0.08

Table 5: Basic descriptive statistics about documents. The length of the document does not include HTML tags.

variable	mean	variance	corr	miss
doc_length	837	1.2×10^3	0.04	0.05
relevance_score	0.49	0.42	0.37	0.18
readability_score	0.52	0.16	0.25	0.11

news source information (RSS_info =<*RSS_link*, *host_link*>) and document length ($doc\ len$);

Level 2, 4 nodes: Hidden variables [5], such as *relevant novel authoritative* and *readable*, that may affect a user's preference for a document;

Level 3, 2 nodes: System generated scores, such as topical *relevance_score* and *readability_score*;

Level 4, 1 node: Whether a user likes a document or not (User judgment of *user_like*);

Level 5, 12 nodes: User actions, such as milliseconds spent on a page (*TimeOnPage*) or the number of clicks on the Down Arrow key (*NumOfDownArrow*).

This informs the learning algorithm that a causation (indicated by →)[6] from a higher/later level to lower/earlier level is prohibited. For example, *TimeOnPage* on the highest/latest level (5th level) couldn't be a cause of *doc_len* on level 1. Although the prior knowledge is engineered by the author and is not guaranteed to be true, it may help the structure learning algorithms by using the constraints to make the search space smaller.

Why *user_like* is on a higher level than user actions? We assume how a user likes (or will like) a document only depends on how well the document satisfies the user's information need. It is a hidden variable that exists before the user reads the document. The user takes a series of actions to uncover the truth. We assume the user judgment of *user_like* is the same as the hidden variable. This assumption is commonly used in the information retrieval community when collecting user assessments. Based on this assumption, user actions won't influence *user_like*, thus we put *user_like* before the actions. Similar assumptions are made for other hidden variables on level 2.

The graph structure learned is presented in Figure 4. This graph is a result of user data and human engineered prior knowledge combined. After learning from the data, the structure of the human engineered prior graph has changed greatly, with more than 80% links removed. It is very encouraging to see that the graph looks reasonable. According to Figure 4, whether a document is *novel*, *relevant*, *authoritative*, *readable* and whether a user is familiar with the topic before using the system (*familiar_topic_before*) are direct causes of the user's preference for a document (*user_like*). How familiar with this topic a user is before participating the user study (*familiar_topic_before*) and the number of web links to the news source (*RSS_LINK*) or host (*host_link*) indirectly affect the user's preference for a page through *relevant* and *authoritative*. *relevant*, *authoritative*, *familiar_topic_before* and *host_link* influence a user's actions, such as the number of events on scroll bars (*EventOnScroll*).

Comparing Tables 2 through 5 with Figure 4, some variables are correlated with *user_like*, however, there are no direct links between them and *user_like*. For example, the correlation between *relevance score* and *user_like* is 0.39, while there is no direct link between them. Does Figure 4 contradict Table 5? The answer is "no". Since there is no direct link between them, *relevance_score* is not a *direct* cause/result of *user_like*. The subgraph *user_like* ← *relevant* → *relevance_score* means *relevance score* and *user likes* share a common cause, *relevant*. The correlation between *relevance_score* and *user_like* is due to the indirect causal relationship between them. Similarly, there is no direct link from the node *user_like* to several variables that are correlated with *user_like* (Section 4.2).

[5] We assume users provided accurate judgment about the hidden variables as explicit feedback.

[6] $X \rightarrow Y$ means X is a direct cause of Y.

Figure 4: A user independent causal graphical structure learned using PC algorithm. The learning algorithm begins with the 5 tier prior knowledge. In the causal graph, $X - Y$ means the algorithm cannot tell if X causes Y or if Y causes X. $X \longleftrightarrow Y$ means the algorithm found some problem. The problem may happen because of a latent common cause of X and Y, a chance pattern in a sample, or other violations of assumptions.

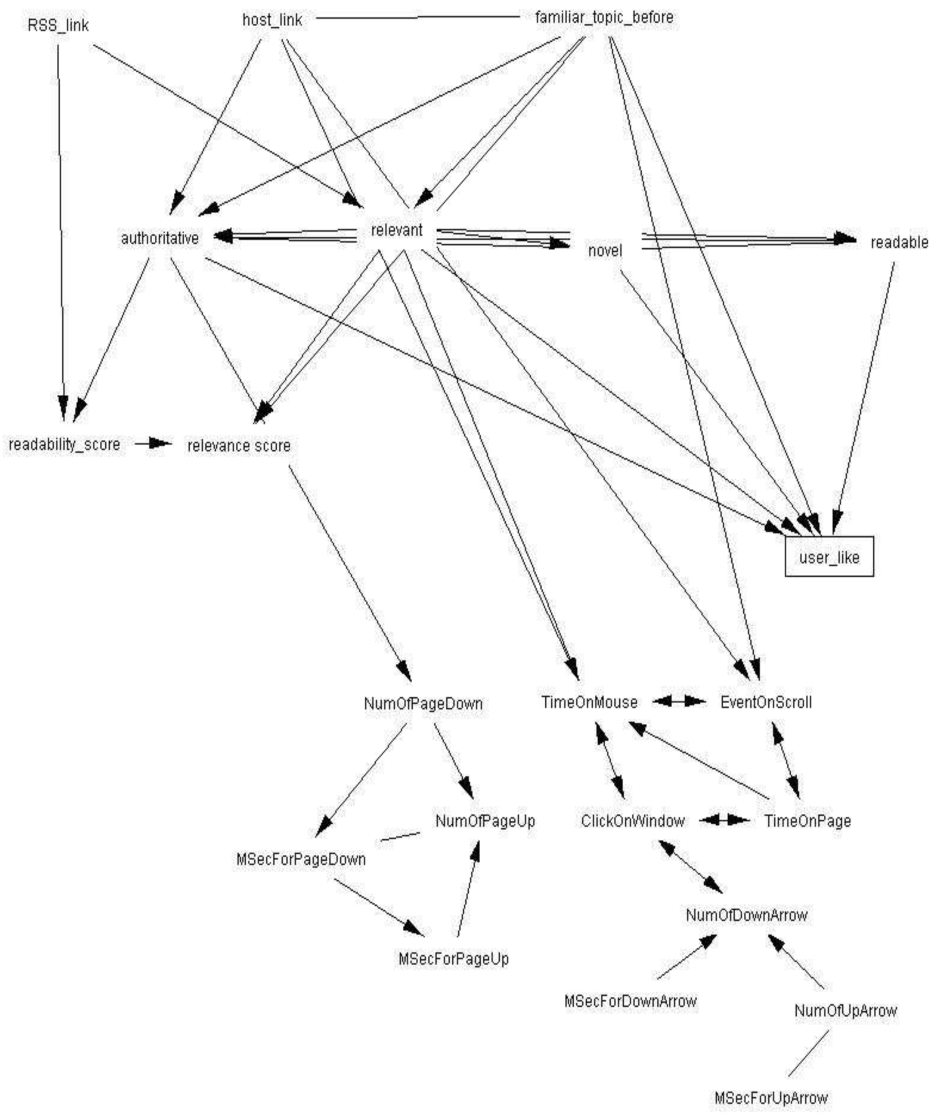


Table 6: The correlation coefficient between explicit feedback.

variable	relevant	novel	authoritative	readable
relevant	1	0.69	0.4328	0.48
novel	0.69	1	0.4381	0.49
authoritative	0.43	0.44	1	0.61
readable	0.48	0.49	0.61	1

The node *user_like* is directly linked to or from *authoritative*, *relevant*, *novel*, *readable* and *familiar_topic_before*. Most of the refined actions, such as the number of times the page up key was pressed (*NumOfPageUp*), are several steps away from *user_like*. This implies that these refined actions are less informative if we want to use the learned model to predict whether a user likes a document or not. This finding agrees with (Claypool *et al.*, 2001) and Table 2.

The node *authoritative* is directly linked to *readability_score* and *host_link*. The link between *host_link* and *authoritative* confirms the existing approaches that use the web link structure to estimate the authority of a page (Kleinberg, 1998). The links between *readability_score*, *readable* and *authoritative* are very interesting. They suggest the difficulty to understand a page may make the user feel it is not authoritative. Further investigation shows that although the percentage of not authoritative news is less than 15% in general, among the 187 news stories some users identified as "difficult" using class labels, 73% were also rated not authoritative. This observation suggests that the estimation of authority of a page may be further improved using the content of a page.

There is a link among the four nodes *relevant*, *novel, readable* and *authoritative*. Further analysis show that the correlation between each pair is high (Table 6). This suggests that the four variables influence each other one way or another. For example, the readability of a document may influence the user's evaluation of authority, while whether a document is relevant or not may influence a user's evaluation of novelty. There are two possible explanations: 1) This is an inherent property of the document; or 2) A user is likely to rate one aspect of the document higher than he should if the other aspects are good.

The link between *readable* and *readability_score* is counter intuitive. To understand why it exists, we need to be aware that the causal relationships learned automatically are what the algorithm believes based on the evidence of the data, the assumptions it makes, and the prior constraints we engineered. The relationships learned may contain error because the data is noisy, or the assumptions and the prior constraints may be wrong. For example, the PC algorithm assumes no hidden variables. However, in additional to *relevant, novel, authoritative, and readable*, other hidden variables, such as whether a document is up-to-date, interesting, misleading, may exist and influence a user's preference for a document (Schamber & Bateman, 1996). For another example, if the ratings influence each other, the prior constraints are no longer true. Instead of uncovering the whole truth, the causal model learned merely shed some light on the relationships between the variables. It only serves as a starting point for us, and more work is needed to better uncover the true. We delay the discussion of future work to Section 5.

Figure 5: Structure of GM_complete.

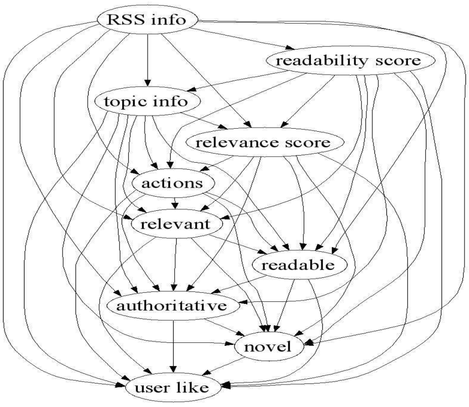
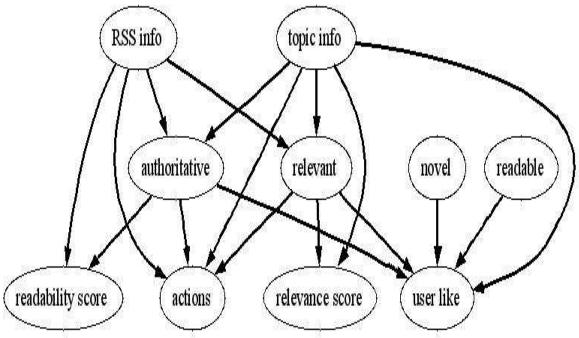


Figure 6: Structure of GM_causal.



4.4 Improving System Performance Using Inference Algorithms

To tell whether combining multiple forms of evidence using graphical models can improve system performance, we evaluated the proposed solution on the task of predicting *user_like* for a given a document. A reliable prediction helps the system decide whether the document should be delivered to the user.

To predict *user_like*, the system learned a graphical model: the combination of a graph structure and a set of local conditional probability functions or potential functions. Doing inference over the causal structure learned in the previous section is difficult because of the cycles and a mixture of directed and undirected links. So, we tried the following directed acyclic graphical models.

GM_complete an almost complete Gaussian network. In this graph, we order the nodes from top to bottom, and the parents of a node were all the nodes above it (Figure 5). Although the nodes were ordered in this figure, one can prove that the actual order of the nodes does not matter. Because learning a joint distribution of all random variables on $GM\_complete$ is equivalent to learning a multi-variate Gaussian distribution without any conditional independence constraints. The learned joint distributions and the final prediction of the model will be the same for different ordering of the nodes.

GM_causal A graphical model inspired by causal models. We manually modify the causal structure in Figure 4 to make it a directed acyclic graph as in Figure 6. The manual process is not optimal and the created model does not strictly maintain all the conditional independence relationship found by the PC algorithm though.

In the graphs, *RSS_info=(RSS_link, host_link)* and *topic_info= (topic_familiar_before, topic_like)* are 2 dimensional vectors representing the information about the news source and the topic in Table 4 and Table 3. $actions = (TimeOnPage, ...)$ is a 12 dimensional vector representing the user actions in Table 1. *user_like* is the target variable the system wants to predict.

To make the probabilistic inference over the graphical models simple, we learned a special family of graphical models, Gaussian networks. If the parents of node X are Y, $P(X|Y) = N(m + W \times Y, \Sigma)$, where $N(\mu, \Sigma)$ is a gaussian distribution with mean μ and covariance Σ. Using the BNT Toolbox (Murphy, 2001), the maximum likelihood estimations of the parameters (m, W, Σ) were learned using the EM algorithm and junction tree inference engine(Cowell *et al.*, 1999) over the graphical models.

Baseline: We used a norm 2 regularized linear regression algorithm as our baseline. We chose this algorithm because of two major reasons. First, other researchers have compared this approach with several state of the art algorithms (eg., logistic regression and SVM) and found it works well (Zhang & Yang, 2003). Second, linear regression is equivalent to the maximum likelihood estimation of a conditional Gaussian model (Hastie *et al.*, 2001), which assumes the conditional probability distribution $P(user\, likes|othervariables)$ is a Gaussian distribution. This assumption is very similar to that of the Gaussian network, thus the major difference between LR models and the graphical models is due to the structure instead of the functional form.

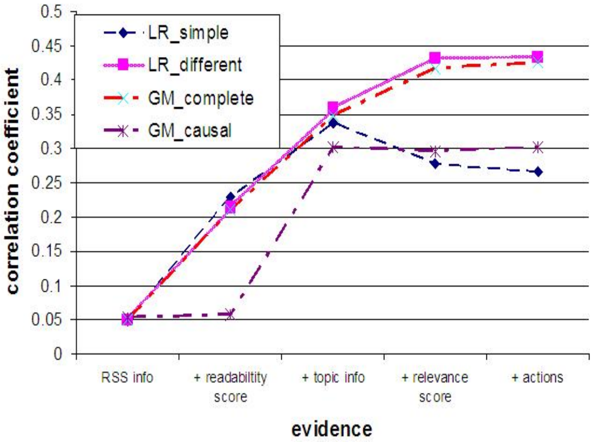
We tried two special approaches to solve the missing evidence problem while using linear regression. The first approach builds a model that does not use the evidence that is missing for each missing situation (*LR_different*). The second approach, *mean substitution*, replaces each missing value for an evidence with the average of the observed evidence (*LR_mean*). For K different forms of evidence, the system may need to handle 2^K different evidence missing situations. A large number of linear regression models would have needed to be learned if we used the first approach, since K is higher than 15 in some of our experiments. Building 2^{15} models is almost impossible for us, so a heuristic approach, which is discussed later, was used to make the experiments possible.

Not all 7,991 cases collected in the user study were used in the experiments. We conducted two sets of experiments. For the first set of experiments, we used the 7,952 cases for which *user_like* was not missing. For the other set of runs, we used only the cases without missing values. In this task, the value of each variable is treated as a continuous value and is normalized to unit variance. Each model was learned from all information available on the first $2/3$ cases, and tested on the remaining $1/3$ cases.

4.4.1 Evaluation Measures

The correlation coefficient between the predicted value of *user_like* and the explicit *user_like* feedback provided by the users was used as the evaluation measure. One baseline is using *relevance_score* alone, which has a correlation coefficient of 0.367 with 95% confidence interval 0.33-0.40 on the last 1/3 of the 7,952 cases.

Figure 7: Comparison of the prediction power of different models using 7952 cases for evaluation. The vertical axis is the correlation coefficient between the predicted value of *user_like* using the model and the explicit feedback provided by the users. The order of different forms of evidence was set manually, based on how easy it was to collect.



4.4.2 Experimental Results and Discussions

Figure 7 shows the effectiveness of different models at different testing conditions as indicated by the horizontal axis. From left to right, additional sources of evidence were given when testing. At the very left of the figure (x=*RSS_info*), a model predicted the value of *user_like* only given the value of *RSS_info* at testing time. "+actions" means the user actions on the current document was given alone with the value of *relevance_score*, *readability_score*, *RSS_info*, and *topic_info*. The graphical models and the *LR_mean* model were trained with all evidence/features, and the learned models were independent of the testing conditions. *LR_different* models were only trained with features provided at testing time, so there is one model per testing condition. [7]

The results show that *GM_complete* performs similarly to *LR_different*. This is not surprising. Theoretically, if there are no missing entries in training data, *GM_complete*'s estimation of $P(user\_like|available\_evidence)$ would be the same as that of *LR_different* on a testing case with missing evidence.

Comparing the correlation coefficients under different testing conditions when using *LR_different* or *GM_complete*, we can see that as more forms of evidence are available, the performance improves. If only the news source information of a document (*RSS_info*) is given, all models perform poorly. The *readability_score* improves the system performance significantly. This is nice and interesting, because the evidence is user independent and can be estimated efficiently for each document. The performance keeps improving as *topic_info* and *relevance_score* were added. To collect this data, we need user feedback on previous documents. The performance improvement is not

[7] However, for a specific testing condition, the training data and testing data contained cases where some evidence that was supposed to be available is missing. In these cases, the training data was ignored and not used to learn a *LR_different* model. However, ignoring such kind of cases in testing data makes comparison of different runs unfair. So we used the mean substitution approach to fill the required missing features in testing data while using *LR_different*.

Table 7: A comparison of different models on all data under the *+relevance_score* (+R) and *+action* (+A) conditions. Corr is the correlation coefficient between the predicted value of *user_like* using the model and the true explicit feedback provided by the users. RLO and RUP are the lower and upper bounds for a 95% confidence interval for each coefficient.

Model	Cond.	corr	RLow	RUp
LR_mean	+R	0.2783	0.2426	0.3132
LR_different	+R	0.4372	0.4058	0.4677
GM_complete	+R	0.4247	0.3928	0.4555
GM_causal	+R	0.3078	0.2728	0.342
LR_mean	+A	0.2646	0.2286	0.2998
LR_different	+A	0.4375	0.406	0.4679
GM_complete	+A	0.4315	0.3999	0.4622
GM_causal	+A	0.3086	0.2736	0.3428

very obvious when *actions* were added. This means that given other evidence (*RSS_info*, *topic_info*, *relevance_score* and *readability_score*), the system will not improve its prediction of (*user_like*) much by observing these actions. However, this is only true when we use a model learned for all users and other forms of evidence are available. It does not mean the actions are useless if we learn user specific models, or if other forms of evidence (such as *relevance_score*) are not available. Meanwhile, since these actions are performed while the user is reading the current document, they couldn't be use for recommending the document. Instead, they may be used to predict how user likes the document, which can be for training the user model for recommending future documents or for collaborative document recommendations (Das *et al.*, 2007).

The performances of *LR_mean* and *GM_causal* do not increase monotonically as more forms of evidence are added. They perform much worse than *LR_different* and *GM_complete*. Why does a structure that looks more causally reasonable perform worse than the simple *GM_complete*? We may answer this question better by comparing the underlying assumptions of these algorithms. *GM_complete* only assumes the joint distribution of all variables is multivariate Gaussian. *GM_causal* makes much stronger independence assumptions by removing some links between variables. As mentioned before, the causal relationships learned automatically are not perfect. This may be the cause of the poor performance of *GM_causal*. *LR_mean* also suffers from the strong conditional independent assumptions.

Table 7 reports the performance together with the confidence intervals of all the models under the *+relevance score* and *+actions* conditions. Under both conditions, *GM_complete* and *LR_different* are statistically significantly (t-test with 95% confidence interval) better than the baseline 0.367. *LR_mean* and *GM_causal* are significantly worse. It means using multiple forms of evidence either hurts or helps, depending on how they are used. Further analysis about the *+actions* runs shows that *LR_mean* gave *explicit feedback* too much weight and overlooked other less strong evidence. At testing time, it did not handle the problem of missing explicit feedback well and thus performed poorly. Although *GM_complete* also gave very high weights to explicit feedback, it could infer the missing values based on other available evidence at testing time, thus performed better than *LR_mean*. *LR_different* did not consider explicit feedback for training, thus it did not overlook other forms

Table 8: The performance on 4522 no missing value cases under the +*relevance_score* (+R) and +*action* (+A) conditions.

Model	Cond.	Corr	RLow	RUp
LR_mean	+R	0.13	0.08	0.18
LR_different	+R	0.41	0.37	0.45
GM_complete	+R	0.41	0.37	0.45
GM_causal	+R	0.41	0.375	0.45
LR_mean	+A	0.11	0.061	0.16
LR_different	+A	0.42	0.38	0.46
GM_complete	+A	0.42	0.38	0.46
GM_causal	+A	0.38	0.33	0.42

of evidence, so it suffered less from the problem. *LR_mean* may work reasonably well if explicit variables are not included, however missing strong evidence will still hurt the performance of *LR_mean* to some extent when there is a large variance on how informative each piece of evidence. For the *GM_complete* approach, a single model is needed to handle various evidence missing situations. If we use the *LR_different* approach, several models are needed. As we mentioned before, there are 2^K different evidence missing combinations, and 2^K linear regression models are needed in order to handle all these situations using *LR_different* approach. *LR_different* may be preferred if K is small, while graphical modelling using *GM_complete* may be a better approach to handle different data missing situations when K is large.

So far, all results are based on 7,952 cases where some evidence may be missing. We also compared the models under different testing conditions using the 4,522 cases that do not have any missing values (Table 8). *GM_causal* performed significantly better than before. We need to be very careful with the structures while using the graphical modelling approach, since a structure that looks more reasonable may work poorly on the inference task. However, we could not draw any conclusion on whether *GM_complete* is better in general, because the answer may be different with different conditional probability distributions, different data sets, or a better structure learning algorithm.

5 Summary

This paper describes a user study to collect an evaluation data for further research on building complex user models for recommender systems. It demonstrates that we can build a longer-term learning environment and collect a significant amount of data about a user's interests with reasonably small effort [8]. We have collected a new evaluation data set that contains thousands of extensive implicit user feedback (such as a user's mouse usage, keyboard usages, and document length), explicit user feedback (such as novel, relevant, readable, authoritative, and whether a user likes a document or not), and other forms of evidence (such as news source information). The basic characteristics, such as the means, for the multiple forms of evidence collected are very diverse. Most forms of evidence are correlated with *user_like*. The correlation between implicit feedback and *user_like* is much weaker than

[8] The author spent about 1 month on designing the user study, 1-2 months on implementing and testing the whole system, and 1 month on running the user study with the real users.

that between explicit feedback and *user_like*. Compared with data collected by other researchers, this data set appears reasonable.

In general, the user study represented a real-world task in a more realistic setting, where users choose to create their own classes and read news using their own computers at the time and place they want. This realistic setting enables us to collect a very detailed filtering data set with ordinary people and available tools. The data is very diverse and possibly more powerful than existing filtering data sets created by NIST for TREC.

The data set is noisy, with many missing entries, and without thorough evaluation for all <document, user class, time> tuples. It would be relatively easy to create a cleaner data set later, but some of the characteristics, such as missing entries and diversity of variables, are common in the real world and unlikely to be eliminated entirely from operational system used by ordinary people.

We have analyzed the user study data using graphical models. The experimental results show that the graphical modelling approach can help us to understand the causal relationships between multiple forms of evidence in the domain and explain the real world scenario better. The results also show that the filtering system can predict user preferences more accurately with multiple forms of evidence than with a relevance model only.

In particular, we studied two problems that are important for adaptive filtering as well as user centered information retrieval based on the graphical modelling approach. First, we studied the **complex user criteria beyond topical relevance**. More specifically, we have developed probabilistic user models with *user_like* and other criteria as hidden variables. We have demonstrated how to quantify the importance of various criteria and combine these criteria with implicit and explicit user feedback based on probabilistic reasoning. The work enables the system to go beyond relevance and develop more interesting and detailed data driven user models than prior research. This is partly because the framework has a better theory, and partly because the advantages of the proposed framework matches the task, where it is practical to collect enough training data to learn over a period of time.

Second, we explored how to solve the **missing data problem** faced by practical recommendation systems. Using more forms of evidence improves the system performance. However, as more forms of evidence are added, missing data is a common problem because of system glitches or because the users will not behave as desired. A real system needs to handle various missing data by either ignoring it or by estimating it based on what is known. The graphical modelling approach addresses this problem naturally. Simpler approaches, such as linear regression, were not designed to handle missing values. In order to use them to combine multiple forms of evidence, extra handling of missing data is needed. *LR_different* handles the problem by building many different models to be used at different data missing conditions. *LR_different* and *GM_complete* perform similarly. When there are few types of evidence, *LR_different* probably is preferable because of the simplicity. However, as more forms of evidence are added, a more powerful model, such as *GM_complete*, may be preferable because of the computation and space efficiency.

In this paper, the system uses a user independent model to combine multiple forms of evidence and learns user dependent models to calculate some types of evidence, such as relevance scores. The major computation of the system was to learn user specific models, such as a relevance model. This means the computational complexity of using graphical models in this paper is similar to that of traditional filtering system. However, if we need to learn a user specific graphical model for inference, the complexity would be higher.

It is worth mentioning that all the inference tasks only considered documents that users clicked and assigned class labels. The performance may be different on arbitrary <document, user class, time> tuples in a real system. A practical filtering system may ask users to create classes manually, or automatically create user classes and assign documents to classes.

We collected data only for documents clicked, and the performance of the algorithms may be different on a random sampled data set with both clicked and un-clicked documents. Further investigation to look at un clicked data is needed, which is a critical step to see whether the findings under the experimental setting described in this paper will help the system serve the user better in a real filtering environment.

This is the first step towards using the graphical modelling approach to build complex user models. The graphical modelling approach is a flexible framework. The proposed solution, especially the data analyzing methodology used in this paper, can also be used in other IR tasks where a rich user profile may help, such as context-based retrieval.

The research reported in this paper is far from the best and there is much room to improve. To mine the real cause-effects relationship underlying the filtering problem, an important future work is to iteratively use different techniques, validate recovered models, and add additional prior knowledge. To improve the prediction power of the modelling approach, an important future work is to compare different graphical models systematically. First, the prediction performance of the filtering system may be improved by adding model selection, variable selection, supervised variables discretization techniques. Second, PC algorithm returns a local minimum in the space of equivalent Bayesian Network, and how to do inference based on the result is an interesting research problem. Based on Figure 4, other causal models for inference should also be considered besides the *GM_causal* used in this paper. We can add links between explicit feedback variables or add links between *RSS_info* and *Topic_info*. Third, the PC algorithm used in this paper is designed to uncover the causal relationships. There are some other structure learning algorithms specially designed for optimizing prediction, such as Bayesian Network Classifiers discussed in (Friedman *et al.*, 1997). Comparing these different graphical models is a future work. Fourth, the missing not at random problem is not directly addressed in this paper, and it would be interesting to try structure learning algorithms, such as (Friedman, 1998), whose hypothesis fits better for this scenario.

6 Acknowledgment

This research was funded by National Science Foundation IIS-0713111, AFOSR/AFRL and the industry sponsors of the Information Retrieval and Knowledge Management Lab at University of California Santa Cruz. Any opinions, findings, conclusions or recommendations expressed in this paper are the author's, and do not necessarily reflect those of the sponsors.

References

Anderson, Corin R., & Horvitz, Eric. 2002. SWeb Montage: A Dynamic Personalized Start Pags. *Pages 704–712 of: roceedings of the 11th World Wide Web Conference*.

Ardissono, Liliana, Console, Luca, & Torre, Ilaria. 2001 (September). An adaptive system for the personalized access to news. *Pages 129–147 of: AI Communications*.

Bharat, K. 2000. Searchpad: Explicit capture of search context to support web search. *Pages 493–501 of: Proceeding of 9th International WWW Conference*.

Billsus, Daniel, & Pazzani, Michael J. 1999. A personal news agent that talks, learns and explains. *Pages 268–275 of: AGENTS '99: Proceedings of the third annual conference on Autonomous Agents*. ACM Press.

Callan, Jamie, Smeaton, Alan, Beaulieu, Micheline, Borlund, Pia, Brusilovsky, Peter, Chalmers, Matthew, Lynch, Clifford, Riedl, John, Smyth, Barry, Straccia, Umberto, & Toms, Elaine. 2003. Personalisation and Recommender Systems in Digital Libraries, Joint NSF-EU DELOS Working Group Report.

Carbonell, Jaime, & Goldstein, Jade. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. *Pages 335–336 of: Proceedings of the 21st annual international ACM SIGIR conference*.

Carreira, Ricardo, Crato, Jaime M., Goncalves, Daniel, & Jorge, Joaquim A. 2004. Evaluating adaptive user profiles for news classification. *Pages 206–212 of: IUI '04: Proceedings of the 9th international conference on Intelligent user interface*. ACM Press.

Claypool, Mark, Le, Phong, Wased, Makoto, & Brown, David. 2001. Implicit interest indicators. *Pages 33–40 of: Intelligent User Interfaces*.

Conati, C., Gertner, A. S., VanLehn, K., & Druzdzel, M. J. 1997. On-Line Student Modeling for Coached Problem Solving Using Bayesian Networks. *Pages 231–242 of: Proceedings of the Sixth International Conference on User Modeling*.

Cowell, Robert G., Dawid, A. Philip, Lauritzen, Steffen L., & Spiegelhalter, David J. 1999. *Probabilistic Networks and Expert Systems*. Springer.

Croft, Bruce, Allan, James, Fisher, David, Strohman, Trevor, Feng, Fangfang, Larkey, Leah, Callan, Jamie, Lafferty, John, Avrahami, Thi Truong, Yau, Lawrence, Ogilvie, Paul, Si, Luo, Collins-Thompson, Kevyn, Turtle, Howard, & Zhai, Chengxiang. 2004. *The Lemur Toolkit for Language Modeling and Information Retrieval*. http://www-2.cs.cmu.edu/ lemur/ (visited Oct. 2006).

Das, Abhinandan, Datar, Mayur, Garg, Ashutosh, & Rajaram, Shyam. 2007. Google News Personalization: Scalable Online Collaborative Filtering. *In: Proc. 16th International World Wide Web Conference*.

Domingue, J., & Scott, P. 1998. KMi Planet: A Web Based News Server. *Page 324 of: APCHI '98: Proceedings of the Third Asian Pacific Computer and Human Interaction*. IEEE Computer Society.

Fox, Steve, Karnawat, Kuldeep, Mydland, Mark, Dumais, Susan, & White, Thomas. 2005. Evaluating implicit measures to improve web search. *Pages 147–168 of: ACM Trans. Information Systems*, vol. 23. New York, NY, USA: ACM Press.

Friedman, Nir. 1998. The Bayesian Structural EM Algorithm. *Pages 129–138 of: UAI*.

Friedman, Nir, Geiger, Dan, & Goldszmidt, Moises. 1997. Bayesian Network Classifiers. *Machine Learning*, **29**(2-3), 131–163.

Harman, Donna. 2003. Overview of the TREC 2002 Novelty Track. *Pages 46–56 of: The Eleventh Text REtrieval Conference (TREC-11)*. NIST 500-251.

Hastie, Trevor, Tibshirani, Robert, & Friedman, Jerome. 2001. *Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer-Verlag.

Heckerman, David, Chickering, David Maxwell, Meek, Christopher, Rounthwaite, Robert, & Kadie, Carl. 2000. Dependancy Networks for Inference, Collaborative Filtering and Data Visualization. *Journal of Machine Learning Research I*, 49–75.

Henzinger, Monika, Chang, Bay-Wei, Milch, Brian, & Brin, Sergey. 2003. Query-free news search. *Pages 1–10 of: WWW '03: Proceedings of the twelfth international conference on World Wide Web*. ACM Press.

Horvitz, Eric, Breese, Jack, Heckerman, David, Hovel, David, & Rommelse, Koos. 1998. The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Soft. *Pages 256–26 of: Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*. San Francisco, CA: Morgan Kaufmann.

Jaakkola, T. S., & Jordan, M. I. 2000. Bayesian logistic regression: a variational approach. *Pages 25–37 of: Statistics and Computing*.

Jordan, Michael I., Ghahramani, Zoubin, Jaakkola, Tommi S., & Saul, Lawrence K. 1999. *An introduction to variational methods for graphical models*. Cambridge, MA, USA: MIT Press. Pages 105–161.

Kelly, Diane, & Teevan, Jaime. 2003. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, **37**(2), 18–28.

Kleinberg, J. 1998. Authoritative sources in a hyperlinked environment. *In: Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*.

Lai, Hung-Jen, Liang, Ting-Peng, & Ku, Y. C. 2003. Customized Internet news services based on customer profiles. *Pages 225–229 of: ICEC '03: Proceedings of the 5th international conference on Electronic commerce*. ACM Press.

Lang, Ken. 1995. NewsWeeder: Learning to filter news. *Pages 331–339 of: Proceedings of the Twelfth International Conference on Machine Learning*.

Le, Phong, & Waseda, Makoto. visited Oct. 2006. *A Curious Browser: Implicit Ratings*. http://www.cs.wpi.edu/ claypool/mqp/iii/.

Mackay, D.J.C. 1998. *Learning in Graphical Models*. MIT Press. Chap. Introducion to Monte Carlo Methods, pages 175–204.

McKim, Vaughn R., & Turner, Stephen (eds). 1997. *Causality in Crisis?: Statistical Methods and the Search for Causal Knowledge in the Social Science*. University of Notre Dame Press.

Merialdo, Bernard, Lee, Kyung Tak, Luparello, Dario, & Roudaire, Jeremie. 1999. Automatic construction of personalized TV news programs. *Pages 323–331 of: MULTIMEDIA '99: Proceedings of the seventh ACM international conference on Multimedia (Part 1)*. ACM Press.

Minka, Thomas P. 2001 (Jan.). *A family of algorithms for approximate Bayesian inference*. Ph.D. thesis, Massachusetts Institute of Technology.

Morita, Masahiro, & Shinoda, Yoichi. 1994. Information filtering based on user behavior analysis and best match text retrieval. *Pages 272–281 of: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc.

Murphy, Kevyn. 2001. *The Bayes Net Toolbox for Matlab*. http://bnt.sourceforge.net/ (visited Oct. 2006).

Pearl, Judea. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc.

Pearl, Judea. 2000. *Causality: Models, Reasoning and Inference*. Cambridge University Press.

Pilgrim, Mark. 2002. *What is RSS*. http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html (visited Oct. 2006).

Pynadath, D.V., & Wellman, W.P. 1995. Accounting for Context in Plan Recognition, with Application to Traffic Monitoring. *Pages 472–481 of: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*.

Schamber, Linda, & Bateman, Judy. 1996 (October). User Criteria in Relevance Evaluation: Toward Development of a Measurement Scale. *Pages 218 –225 of: ASIS 1996 Annual Conference Proceedings*.

Shen, Xuehua, Tan, Bin, & Zhai, ChengXiang. 2005. Context-sensitive information retrieval using implicit feedback. *Pages 43–50 of: SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM Press.

Spirtes, Perter, Glymour, Clark, & Scheines, Richard. 2000. *Causation, Prediction, and Search*. The MIT Press.

Sugiyama, Kazunari, Hatano, Kenji, & Yoshikawa, Masatoshi. 2004. Adaptive web search based on user profile constructed without any effort from users. *Pages 675–684 of: WWW '04: Proceedings of the 13th international conference on World Wide Web*. ACM Press.

Tanner, Martin A. 1996. *Tools for Statistical Inference*. 3 edn. Springer.

Thomas, A., Spiegelhalter, D.J., & Gilks., W.R. 1992. BUGS: A program to perform Bayesian inference using Gibbs sampling. *Pages 837–842 of: Bayesian Statistics 4*.

Varian, H. R. 1999. *Economics and search (Invited talk at SIGIR 1999)*.

Voorhees, E. M., & Buckland, Lori P. (eds). 2002. *NIST Special Publication 500-251: The Eleventh Text REtrieval Conference (TREC 2002)*. Department of Commerce, National Institute of Standards and Technology.

Wang, Peiling. 1994. *A cognitive model of document selection of real users of IR Systems*. Ph.D. thesis, University of Maryland.

White, Ryen W., Jose, Joemon M., & Ruthven, Ian. 2006. An implicit feedback approach for interactive information retrieval. vol. 42. Tarrytown, NY, USA: Pergamon Press, Inc.

Yedidia, J.S., Freeman, W.T., & Weiss, Y. 2000. Generalized Belief Propagation. *Pages 689–695 of: Advances in Neural Information Processing Systems (NIPS)*, vol. 13. National Institute of Standards and Technology, special publication 500-249.

Zhai, C., Cohen, W., & Lafferty, J. 2003. *Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval*.

Zhang, J., & Yang, Y. 2003. Robustness of regularized linear classification methods in text categorization. *In: ACM SIGIR'03*.

Zhang, Yi. 2004. Using Bayesian Priors to Combine Classifiers for Adaptive Filtering. *In: Proceedings of the 27th Annual International ACM SIGIR Conference*.

Zhang, Yi, Callan, Jamie, & Minka, Thomas. 2002. Novelty and redundancy detection in adaptive filtering. *Pages 81–88 of: SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM Press.