A Graph-based Recommender System for Digital Library

Zan Huang, Wingyan Chung, Thian-Huat Ong, Hsinchun Chen
Artificial Intelligence Lab
Department of Management Information Systems
The University of Arizona
Tucson, Arizona 85721, USA
1-520-621-2748
{zhuang, wchung, tong, hchen}@eller.arizona.edu

ABSTRACT

Research shows that recommendations comprise a valuable service for users of a digital library [11]. While most existing recommender systems rely either on a content-based approach or a collaborative approach to make recommendations, there is potential to improve recommendation quality by using a combination of both approaches (a hybrid approach). In this paper, we report how we tested the idea of using a graph-based recommender system that naturally combines the content-based and collaborative approaches. Due to the similarity between our problem and a concept retrieval task, a Hopfield net algorithm was used to exploit high-degree book-book, user-user and book-user associations. Sample hold-out testing and preliminary subject testing were conducted to evaluate the system, by which it was found that the system gained improvement with respect to both precision and recall by combining content-based and collaborative approaches. However, no significant improvement was observed by exploiting high-degree associations.

Categories and Subject Descriptors

H.3.7 [**Digital Libraries**]: Systems Issues.

General Terms: Algorithms, Design, Experimentation.

Keywords: Recommender system, Hopfield net algorithm, Graph-based model, Content-based filtering, Collaborative filtering, Mutual Information algorithm, Chinese phrase extraction

1. INTRODUCTION

The over abundance of document information in digital libraries has created much inconvenience to users seeking documents online. Users with diverse backgrounds and interests are all provided the same information in response to the same query terms. Individual histories of library usage are not used in suggesting books to users. To retrieve relevant documents upon a keyword search, existing digital libraries exclusively rely on document content information to make recommendation. Consequently, although digital libraries contain rich information of users' usage history, demographic and book content information, much of it is not used to provide value-added services to users

A similar situation appears in the commercial world, where the over abundance of product information provided by online stores prevents customers from searching products efficiently. Too much product information is provided and much of this information is not relevant to a specific customer. Some online stores employ various approaches and information sources to suggest relevant products to customers. Among them the best known example is Amazon.com, which makes personalized recommendations to its customers using combination of collaborative recommendation based on purchase history and customer ratings, manual recommendations and customer text comments [22]. As a new category of software [20], recommender systems can be also applied to digital libraries. In this paper, we developed a graph-based model that combines content-based and collaborative recommendation approaches and implemented the system in the context of an online Chinese bookstore. In order to process Chinese text, Chinese phrase extraction with Mutual Information algorithm is one of the core components in the system. The online bookstore records the book content information, customers' demographic information and their purchase histories, which are similar to document content information, users' personal attributes and their usage history, respectively, in a digital library environment.

2. LITERATURE REVIEW

Recommender systems advise users on relevant products and information by predicting a user's interest in a product, based on various types of information such as users' past purchases and product features. Schafer, Konstan and Riedl [23] proposed a taxonomy for recommender applications in the context of electronic commerce that is also appropriate for general recommender systems. For any recommender system, the choices of information sources for the system and the quality of the information sources are vital to success. Schafer et al. categorized recommender systems

JCDL'02, July 13-17, 2002, Portland, Oregon, USA.
Copyright 2002 ACM 1-58113-513-0/02/0007…$5.00.

according to the information sources utilized by the system, which they described as “community inputs”. These include item attribute, external item popularity, purchase history, ratings, text comments and context/process information. For digital libraries, all these information can be conveniently obtained by substituting borrowing history for the purchase history.

Another important characteristic of recommender systems is the recommendation methods it utilizes. Schafer et al. listed 6 categories as most current recommendation systems: raw retrieval, manual selection, statistical summarization, attribute-based, item-to-item correlation, and user-to-user correlation. This paper will focus on the last two methods.

A recommender system using the item-to-item correlation method recommends products to a customer based on associations between the new item and items in which the customer has expressed interests. This method is also known as “content-based approach”. An item profile usually consists of some attributes of the product, and the user profile is created from the product profiles involved in the historical behavior. Some similarity function is used to match profiles to give recommendations. Examples of recommender systems using this method are reported in [24] and [12].

Many other recommender systems use the user-to-user correlation method, also known as the “collaborative filtering approach”. These systems give recommendations based on correlation between users observed by looking at user behaviors in the system. A customer profile is usually created on the basis of his/her past behavior, and similarity functions are used to find similar customers. This approach addresses the phenomenon that when selecting products or information, people are usually influenced by other people’s experiences with the product or information. Systems like those in [19], [17] are examples using this method.

The content-based and the collaborative filtering approaches are not mutually exclusive to each other, and there have been many efforts to integrate them in order to obtain more accurate recommendations. These systems can be loosely categorized into several classes. A simple way to combine the content-based and collaborative-filtering recommendation approaches is to perform recommendation using the two approaches separately and combining the results. An example system in this category is the Profbuilder system, in which users are provided from both lists content-based and collaborative filters [1]. Claypool et al. reported their efforts actually to combine the two recommendation lists, and the system is personalized in the sense that the combination weights of the content-based and collaborative predictions are determined on a per-user basis. [8] These systems do not truly use a hybrid approach, because the processes of content-based and collaborative recommendation are kept separate.

Some more “hybrid” systems combine the two approaches at a lower level by generating hybrid representations of items or users. For a recommender system, three types of information are usually available: factual information of the item, user demographic information, and transactional information. In domains like books, news, and movies, item factual data usually contains text descriptions of product names, specifications, characteristics, and etc. The most commonly available transactional information is users’ ratings and purchase history. A pure collaborative filtering approach usually relies only on user demographic information and transactional information, while a content-based approach usually utilizes item factual data. Systems using hybrid approaches attempt to combine different information sources at the representation level. Some systems incorporate content-based filtering results into transactional information, as does GroupLens system, in which filterbots are treated as normal users and new articles are evaluated using content-based filtering approaches [22]. The filtering results are transformed into ratings from filterbots and become part of the user-article transactional information. Another example is the Fab system, in which content information of items associated with users in the transactional data is part of the user representation and content-based approaches are used to formulate user similarities in a collaborative recommendation [3]. This approach is also called “collaboration via content” [19]. Other systems incorporate transactional information into the representation of users as well as items. In the system reported by Basu, Hirsh and Cohen, users are represented by a set of items and items are represented by a set of users, and inductive learning method is used to predict user’s movie preferences based on the user-movie pair [5].

The hybrid recommendation systems listed above use different approaches and representations to fully utilize the product, customer and transaction information available. Usually significant human engineering and ad hoc design efforts are needed to incorporate information into appropriate representations. Some other researchers did not attempt to incorporate different information sources into the book or user representation but proposed models containing all information sources and then apply inductive learning techniques to mine preference recommendations. Condliff, Lewis, Madigan and Posse reported a universal model for inductive learning in which users and items are represented by feature vectors, and transactional user-item interaction are represented as training data [9]. Any inductive learning or statistical method can be applied in this model, and in their work Bayesian mixed-effect model was used.

These hybrid systems reported different degrees of prediction accuracy gain from utilizing multiple information sources, ranging from modest benefit [9] to significant improvement [5]. Basu et al. also reported that adding information does not always lead to monotonically better results. [4]

3. MODEL AND CONTEXT: A TWO-LAYER GRAPH APPROACH

In this paper we describe a generic, graph-based recommendation approach to integrate the content-based approach with the collaborative-filtering approach in the context of digital libraries by representing books and users in an extended graph that incorporates book-to-book correlation, user-to-user correlation and book-to-user correlation. We use a dataset obtained from a major Chinese online bookstore in Taiwan as our exploratory domain because the application is generic and the characteristics are similar to digital libraries.

Data covering the purchase and book information of five recent years consist of three types of information: books, user demographics and orders. A total of 9,695 books, 2,000 customers, and 18,771 transactions are included in the dataset and assigned to the categories in Figure 1.

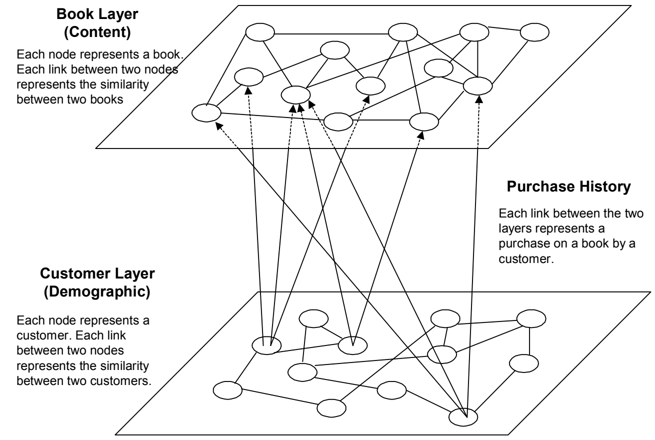
FIGURE 1. Fields of each table

Books		Demographics	Orders
Item number	Translator	Customer ID	Customer ID
Chinese title	Original author	Country	Item number
Foreword	Keyword	City	List price
Table of content	Publication date	Birthday (month)	Order time
Introduction	Cover	Birthday (date)	Payment type
Author	Size	Birthday (year)	Shipping method
Award	Pages	Education	
Category	Weight	Vocation	
Original title	Printing	Sex	
ISBN	Layout	If_Married	
Publisher	Language	If_have_Child	
Series		Number of children	
Edition		Average age of children	
9695 records		2000 records	18771 records

Our approach comprises of two stages of computation. In the first stage, we represent the customers and books by feature vectors from factual customer and book information, which is similar to the model in Condliff et al.'s work. The feature vector for each customer contains the customer demographic data and the book feature vector contains both attributes of the book and text information about the books coming from the title, introduction, foreword, and etc. Unlike Condliff et al.'s model, where the feature vectors were fed directly into the recommendation model, we compute similarity between customers and similarity between books based on the feature vector by applying some similarity functions, which makes the our model simple and intuitive.

At the second stage, we model the books, customers and purchase transactions in an extended graph. Using the similarity weights obtained from the first stage, we construct a two-layer graph (Figure 2) that consists of a book layer and a customer layer. Each node in the book layer represents a book while each link between any two books represents the content similarity between them. Each node in the customer layer represents a customer, and links between customer nodes are the demographic similarity between the two customers. Besides the inner layer links, inter-layer links are also available in the graph model. These links are based on the purchase histories of all customers, wherein a purchase is represented by a link between a node in the book layer and a node in the customer layer.

FIGURE 2. Two-layer graph model of books, customers and purchases


Book Layer (Content)
Each node represents a book. Each link between two nodes represents the similarity between two books
Purchase History
Each link between the two layers represents a purchase on a book by a customer.
Customer Layer (Demographic)
Each node represents a customer. Each link between two nodes represents the similarity between two customers.

In our model, the recommendation activity becomes a graph search task. The three types of links in the graph model are traversed to find books that have strong association with the customer. Different graph search methods can be used to identify recommendation items. We have a detailed example to explain the computation of associations between book and customers in the next section.

We believe that this model is flexible, comprehensive and modular. Firstly, the similarity weights computed in the first stage can be flexibly adjusted to reflect the importance of certain aspects of the data. For example, if we want to emphasize the importance of author in our recommendation, we can simply assign a higher weight to it. If we want to recommend based on book content information, we can assign a higher weight to the texts appearing in the introduction, keyword, and topic fields. The flexibility is believed because that we can control the parameters easily without building new models.

Secondly, this model covers all the three approaches of recommendation. The content-based, collaborative and hybrid approaches can be applied in this comprehensive model. If we use only book-to-book similarity weights to make a recommendation, it is purely content-based. If we use only customer-to-customer similarity weights and purchase histories to make recommendation, we are using a purely collaborative approach. We can combine both approaches by using all the association weights and purchase histories, which is considered a hybrid approach. Covering three recommendation approaches in a two-layer graph as this model does is comprehensive.

Thirdly, this model is modular and allows for future expansion. Since the two computation stages described are independent of each other, we can adopt different algorithmic techniques on each stage to test for different performances. For example, we can adjust the way we compute similarity weights in stage one without changing the method we use in stage two. We can also use other graph searching techniques in stage two for better performance. In our preliminary experiment, we used two graph search methods in the second-stage computation. One was a simple graph search that used only low-degree associations between books and customers. The other was a neural network approach using a Hopfield algorithm to achieve spreading activation and exploit higher degree associations between books and customers.

4. REPRESENTATION AND ALGORITHM DETAILS

This section describes details of representing books, extracting Chinese key terms or phrases from book content information, computing book-to-book similarity weights and customer-to-customer similarity weights, and performing Hopfield net spreading activation.

4.1 Book Representation

Each book contains two types of information, namely, book content information and book attributes information. Book content information comes from the fields of title, keyword, topic, foreword and introduction. Book attributes information comes from the other fields in the book table (Figure 1) such as number of pages, book layout, printing, and so on. We represent each book as a combination of content phrase vector and book attributes. The Chinese phrases are extracted using the Mutual Information algorithm described below.

4.2 Mutual Information Key Phrase Extraction

Phrase extraction in Chinese involves finding the longest phrase in a word string that has precise meaning, similar to finding noun phrases in a sequence of English words [15]. Key phrase extraction is an important step for content-based filtering, because these key phrases are used to represent the content of a book.

The mutual information approach is an iterative process of identifying significant lexical patterns by examining the statistical frequencies of word co-occurrences. *Mutual information* is a metric that measures how frequently a pattern happens in the corpus, relative to its sub-patterns. In its simplest form for a pattern *c*, the formula for calculating MI_c can be expressed as follows:

$$MI_c = \frac{f_c}{f_{left} + f_{right} - f_c}$$

where *f* stands for the frequency of a given word or word pair and MI_c represents the mutual information, which is the probability of co-occurrence of the pattern *c* (e.g., 人工智慧 meaning artificial intelligence), relative to its left sub-pattern (人工智) and the right sub-pattern (工智慧). If MI_c is high and close to 1, it means the pattern *c* is more likely to form a phrase than its left and right sub-patterns alone. On the other hand, if MI_c is low and close to 0, pattern *c* is not likely to form a phrase.

An important algorithmic requirement for this approach to work is an efficient data structure that makes it feasible to analyze a large collection of training corpuses [18]. However, a greater challenge is finding all possible patterns that exist in the corpus along with their frequency occurrence in the corpus. Chien's PAT-tree approach solved the problem in Chinese with relative ease, because PAT-tree is highly efficient and well suited for accessing a large corpus. In addition, previous research in text searching further suggests that other similar data structures including suffix array or PAT-array could be equally efficient [2] [16].

The Mutual information approach could be seen as an extension to bi-gram, tri-gram and even n-gram in traditional information retrieval techniques. Kenneth Church pointed out that some Bible literature has repeated patterns with up to 400 words [7], making it a challenge for the n-gram technique, because without removal, every sub-pattern in the 400-word sentence will be extracted out. Removal of a pattern from the corpus has an effect to the frequency distribution of the corpus, especially repetitive removals of many sub-patterns. To solve this problem, we use a new data structure we developed earlier [18] to support consistent on-line frequency update after removing patterns that we have already extracted. This method increases the success of subsequent extractions, because the 400-word sentence will now be extracted only once and its sub-patterns will not be extracted. Although Chien's approach does not call for stop words, our preliminary experience suggests that the use of stop words could increase the accuracy of the results [18].

Based on the Chinese key phrase dictionary created by the mutual information algorithm, five lists of key phrases from title, keyword, topic, foreword and introduction are constructed for each book. Because tagged author information is not available in the original data, we treated author as another content field in the preliminary experiment. Thus we have six lists of key phrases altogether. These are integrated into one key phrases list to represent the book. Each of the key phrases in the representation list is one descriptor of the book. Key phrases from the fields of title, keyword and author are assigned higher weights to represent the books. Based on information retrieval literature, a final content similarity measure is computed using a modified version of asymmetric cluster function that supports key phrases with different weights.

4.3 Similarity Computation

Based on the key phrase vector model [21], the combined weight of descriptor *j* in book *i* is computed by multiplying the *term frequency* and *inverse document frequency* as follows

$$d_{ij} = tf_{ij} \times \log\ [\ (N / df_j) \times w_j\ ] \qquad \text{…… (1)}$$

The modified version of the combined weight function supports phrases (terms) with different weights.

$$d_{ij} = tf_{ij} \times \log\ [(N / df_j) \times w_j] \times pw_j \qquad \text{…… (1')}$$

where pw_j represents the phrase weight of descriptor *j*. A descriptor can be a term or phrase containing any number of Chinese characters.

Since it has been shown that an *asymmetric algorithm* performs better than other clustering function like cosine function [13] to compute similarity for term associations, extending it to books is expected to improve precision as well.

$$\text{Cluster Weight } (B_j, B_k) = \frac{\sum_{i=1}^{n} d_{ijk}}{\sum_{i=1}^{n} d_{ij}} \times \frac{\log(N / df_k)}{\log(N)} \quad \text{…… (2)}$$

$$\text{Cluster Weight } (B_k, B_j) = \frac{\sum_{i=1}^{n} d_{ikj}}{\sum_{i=1}^{n} d_{ik}} \times \frac{\log(N / df_j)}{\log(N)} \quad \text{…… (3)}$$

where d_{ij} is the combined weight of descriptor *j* in book *i* as computed in (1 '),

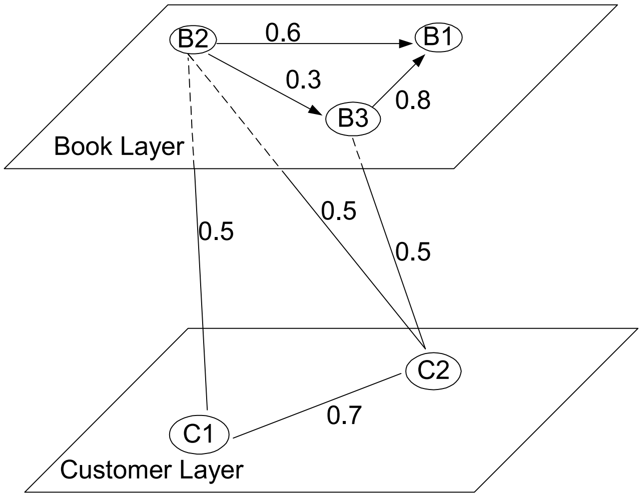
$d_{ijk} = tf_{ijk} \times \log\ [(N / df_{jk}) \times w_j]$ is the combined weight of descriptor *i* in combination of book *j* and *k*, where

- *N* represents the total number of books
- tf_{ij} represents the number of occurrences of descriptor *j* in book *i*
- $tf_{ijk} = tf_{ij}$ when descriptor *i* appears in both book *j* and *k* , otherwise $tf_{ijk} = 0$
- df_j represents the number of books containing descriptor *j*
- df_{jk} represents the number of common descriptors that appear in book *j* and *k*
- w_j represents the number of Chinese characters in descriptor D_j.

The above similarity is asymmetric because the similarity weight from book *i* to book *j* may not be the same as the similarity weight from book *j* to book *i*. In addition to the content similarity measure of books, other book attributes such as publisher, category, layout and so on are used to compute a book attribute similarity weight. Such similarity weight is symmetric because two books can either have or not have the same value for those attributes. For example, if two books are published by the same publisher, then they are similar and the symmetric weight between them should reflect this information. A weighting scheme is used to combine the two similarity measures to form the final book similarity. The customer similarity measure is computed from customer demographic data, which reflect relatively weaker association than book similarity and purchase link, thus a lower weight is given to customer similarity. The inter layer links between book layer and customer layer is simply derived from purchase history. Each link in the graph has a weight between 0 and 1.

After the construction of the two-layer graph, the recommendation activity becomes a graph search task. We use the following simplified example (Figure 3) to illustrate the basic idea of recommendation from graph search.

FIGURE 3. Simplified two-layer graph example



Recommendation suggestion is generated based on the association strength between a customer and a book. In the example in Figure 3, the association between C1 and B1 is to be estimated. If only 1-degree association is allowed, then no association between C1 and B1 can be inferred. In this case, association between customer and book exist only when a customer actually has bought the book, which is not very useful information for a recommendation prediction. When 2-degree association is allowed, the path of C1-B2-B1 can be utilized to form an association estimate between C1 and B1, where the path consist of one purchase link C1-B2 and one book similarity link B2-B1. We compute the association measure as the product of all association weights along the path, and in this case the association measure between C1 and B1 is 0.5*0.6 = 0.3. This result is actually based on a pure content-based approach. When 3-degree association is allowed, there will be 3 more paths from C1 to B1, C1-C2-B2-B1, C1-B2-B3-B1, and C1-C2-B3-B1. By summing up association measures of all of these 4 paths, we get the 3-degree association measure of C1 and B1, which is 0.3 +

0.21 + 0.12 + 0.28 = 0.91. This result is in fact a combination of the content-based and collaborative approaches. In theory, we can go on to increase the degree of association to *n*, but the computation requirement increases exponentially with the increase of the degree of association. The recommended book list for a customer will consist of the books with the highest association measure with the customer and books that have not been previously purchased by the customer.

In our experiment, we evaluated low-degree association and high-degree association recommendation prediction. For low-degree association recommendation, we retrieved books that were similar to the customer's previous purchase as content-based recommendations, and we retrieved the books that had been purchased by the customers who also had bought the books in the target customers' purchase history as collaborative recommendations. For the high degree association recommendation, a Hopfield net algorithm was used to perform spreading activation and obtain recommendations for contend-based, collaborative and hybrid approaches.

4.4 Hopfield Net Algorithm

The high-degree association recommendation can be treated as a concept retrieval task. Based on previous research [6, 13], Hopfield algorithm was found to be suited for concept-based information retrieval. The Hopfield algorithm (or Hopfield Net) performs a parallel relaxation search, in which nodes are activated in parallel and activation values from different sources are combined for each individual node. Neighboring nodes are traversed in order until the activation levels of nodes in the network converge. In the present model, our weighted network of books and customers can be perceived as interconnections of neurons and synapses in the Hopfield net, where neurons represent books or customers and synapses represent weighted links between pairs of books and customers. This algorithm terminates when there is no significant difference in terms of output between two iterations. The following presents a sketch of the Hopfield net activation:

Initialization with User's input: The system first retrieves all the books that were purchased by the customer previously as the starting nodes in iteration 0.

$$\mu_i(0) = x_i,\ 0 \leq i \leq n\text{-}1$$

$\mu_i(t)$ is the weight of node *i* at iteration *t* and x_i, which ranges between 0 and 1, indicates the input weight for node *i*. At iteration 0 ($t = 0$), all starting nodes are assigned weight of 1, all other nodes are assigned weight 0.

Activation, weight computation and iteration: The output of each node is computed as follows

$$\mu_i(t+1) = f_s\left[\sum_{i=0}^{n-1} t_{ij}\mu_i(t)\right],\ 0 \leq j \leq n\text{-}1$$

where t_{ij} equals the association weight between node *i* to node *j* when there is a link that points from node *i* to node *j*, otherwise t_{ij} equals 0.

In order to obtain a result in a reasonable amount of time, we keep the top 50 nodes (ranked by $\mu_i(t)$) as activated nodes for the next iteration. So, $n = 50$. The f_s is the SIGMOID transformation function [10, 14] and is shown below

$$f_s(net_j) = \frac{1}{1+\exp\left[\dfrac{\theta_j - net_j}{\theta_0}\right]}$$

where $net_j = \sum_{i=0}^{n-1} t_{ij}\mu_i(t), \theta_j$ serves as a threshold or bias, and θ_0 is used to modify the shape of the SIGMOID function. θ_j and θ_0 need experiments to be adjusted according to the characteristics of the graph and magnitude of the association weights in the graph.

Criteria for Convergence: The above process is repeated until there is no significant change in output between two iterations, which can be checked by

$$\sum_{j=0}^{n-1} \left|\mu_j(t+1) - \mu_j(t)\right| \leq \varepsilon$$

where ε is the maximal allowable error (a small number). In the final iteration, only the top 50 book nodes that had not been purchased by that customer are retrieved as the list of recommended books.

5. RESEARCH DESIGN AND IMPLEMENTATION

5.1 Research Questions and Hypotheses

Our study addressed two questions: Does a hybrid recommendation approach perform better than a purely content-based approach and a purely collaborative approach? Does high-degree association improve the effectiveness of recommendation results over low-degree association? We used precision and recall as our primary measures of effectiveness.

The specific hypotheses examined in our study were:

- H1: Hybrid recommendation approach achieves higher precision than content-based approach does.
- H2: Hybrid recommendation approach achieves higher precision than collaborative approach does.
- H3: Hybrid recommendation approach achieves higher recall than content-based approach does.
- H4: Hybrid recommendation approach achieves higher recall than collaborative approach does.

- H5: Exploiting high-degree association relationships in the book-customer graph achieves higher precision than only exploiting low-degree association relationships does.
- H6: Exploiting high-degree association relationships in the book-customer graph achieves higher recall than only exploiting low-degree association relationships does.

5.2 System Implementation

The system was mainly implemented in databases (Microsoft SQL Server) on the Windows NT environment. A C++ implementation of Mutual Information Algorithm on Unix environment is used to extract Chinese key phrases. All other algorithms including asymmetric cluster weight function and Hopfield net spreading activation are implemented using SQL Server stored procedures. The graph link formation stage computation is performed off-line; in real applications it can be performed in batch mode when new books, customers and purchase data are added into the database. The graph search stage computation is performed in real time. Customers get their recommendations based on previous purchases or books in which the customer expressed interest at run time. Original book, customer and purchase history data, preprocessed data, and recommendation results are stored in database tables. This leverages on the database management software and at the same time achieves a high degree of reusability.

6. EXPERIMENTS AND EVALUATION

To compare the effectiveness of the different recommendation designs, we conducted a preliminary experiment, which includes two types of evaluations.

6.1 Hold-out Testing

In the evaluation we borrowed ideas from machine learning cross validation methods. We randomly selected a group of 100 customers from the database, and retrieved from the purchase history data a list of books for each customer. Each list is ordered by the purchase date. For each selected customer, more recent purchases (the second half of the list) were treated as "future" purchases and served as target recommendations for that customer. Purchase links based on these "future" purchases were removed from the two-layer graph before generation of recommendations. Earlier purchases (the first half of the list) were used to generate recommendations, and excluded from the recommendation lists. We compared the recommendation list with the "future" purchase list to calculate precision and recall to evaluate different designs. Two-Factor ANOVA tests were conducted to test the effects of degree of associations and recommendation approaches on precision and recall.

Some of the initial experiment and analysis results are listed in Table 1 and Table 2.

TABLE 1. Hold-out Testing Precision and Recall (sample size: 100)

	Low-degree association		High-degree association	
Approach	Precision	Recall	Precision	Recall
Content-based approach	2.48%	10.58%	1.94%	7.80%
Collaborative approach	1.56%	8.15%	1.78%	10.43%
Hybrid approach	**2.76%**	**14.52%**	**3.32%**	**13.51%**

TABLE 2. Two-Factor ANOVA Results for Precision and Recall

Two-Factor ANOVA Results (Precision) alpha: 0.05

Source of Variation	SS	df	MS	F	P-value	F crit
Columns	9.6E-05	1	9.6E-05	0.0306	0.8611	3.857
Rows	0.019049	2	0.00952	3.0404	**0.0486**	3.011
Interaction	0.003172	2	0.00159	0.5063	0.603	3.011
Within	1.86084	594	0.00313			
Total	1.883157	599				

Two-Factor ANOVA Results (Recall) alpha: 0.05

Source of Variation	SS	df	MS	F	P-value	F crit
Columns	0.003789	1	0.0038	0.1086	0.7419	3.8572
Rows	0.304047	2	0.152	4.355	**0.0133**	3.0109
Interaction	0.066006	2	0.033	0.9454	0.3891	3.0109
Within	20.73523	594	0.0349			
Total	21.10907	599				

We conducted an ANOVA test to compare differences between low-degree association and high-degree association methods (columns) and the differences between using content-based, collaborative and hybrid approaches (rows). Through this test we wanted to investigate the impact of these two factors on the precision and recall of the recommendation results. As shown in Table 2, the only two P-values that reflect significant differences are the precision between rows (0.0486) and the recall between rows (0.0133), which reveal that ***there are statistically significant differences with respect to precision and recall among content-based, collaborative and hybrid approach*** and ***no statistically significant difference was found between low-degree association and high-degree association across all three recommendation approaches***. Pairwise t-tests were conducted to further evaluate performance differences between different approaches. It was found that ***hybrid approach outperforms either pure content-based approach*** (t-test p-values, precision: 0.0058; recall: 0.000037) ***or collaborative approach*** (t-test p-values, precision: 0.0016; recall: 0.00017), and ***no significant difference is observed between content-based and collaborative approaches***. In summary, hypotheses H1-H4 were supported by our experiment results and hypotheses H5 and H6 were rejected.

Not surprisingly, the precision and recall of all six-recommendation designs in the hold-out test were very low. It is because we used the first half of customers' purchases to predict their second half purchases, and there was a gap between customer interest and actual purchase behavior. There are many other factors that affect customer's purchase behavior. This may partially explain the low precision and recall.

6.2 Subject Evaluation

In this evaluation we intended to find out human subjects' evaluation of the recommendation system. The subjects were given the purchase histories of selected customers, and asked to use their knowledge about the books, book purchase experience and information from the online bookstore to make recommendations. The two subjects in this experiment were well-educated (with Master's Degrees) Taiwan citizens who had lived there for more than 20 years and could represent typical human recommenders. Their recommendations were used as target recommendations in evaluating the effectiveness of different designs. We generated six recommendation lists from different experiment designs for each of the 3 selected customers, whose historical purchase information was presented to the subjects. We selected specific testing customers such that the historical book purchases were within the interests or domains of knowledge of the subjects. The subjects could form a basic estimation of the customer interests, after which we asked the subjects to recommend books to the customers. A randomly ordered recommendation list integrated from the six recommendation lists was presented to the subjects. Subjects could select more books from these lists as their recommendations. The final recommendation lists given by the subjects were treated as benchmarks to compute precision and recall of different recommendation designs. The results are listed in Table 3.

TABLE 3. Subject Testing Precision and Recall

	Low-degree association		High-degree association	
Approach	Precision	Recall	Precision	Recall
Content-based approach	**16.0%**	**36.3%**	**18.3%**	**38.1%**
Collaborative approach	7.0%	27.7%	6.0%	20.7%
Hybrid approach	10.3%	31.6%	7.0%	34.4%

Based on the results from pilot subject test, the content-based approach outperformed collaborative and hybrid approaches, and no significant benefit of exploiting high-degree association was observed. It was observed that the subjects formulated preferences of customers by identifying major themes in their purchase histories, and then gave recommendations mainly based on these themes. Subjects tended to rely heavily on the book content information, which explains the surprisingly better performance of the content-based approach over the collaborative and hybrid approaches. It is expected that book-selling experts would utilize more comprehensive information to make recommendations to customers than regular customers would, and that experiments conducted with these experts might evaluate the performance of our recommender system more precisely.

Another phenomenon we noticed that may had an impact on our results was that most people in Taiwan still chose traditional bookstore when purchasing books, so the book purchase history data we got from the online bookstore in this research might not have fully represented the customer interests. Incomplete representation of the customers might also have led to unsatisfactory performance of the system.

7. CONCLUSIONS AND FUTURE WORK

In this paper we have described a two-layer graph model in the context of book recommendation that is generic to content-based, collaborative or hybrid recommendation approaches. Using our model a recommendation becomes a graph search activity, and different graph search approaches can be applied. In our preliminary experiment, we compared low-degree association and high-degree association graph search approaches. Hopfield net spreading activation was used for the high-degree association graph search. A hold-out test and a pilot subject test were performed to evaluate the performance of different combinations of recommendation approaches.

Based on the hold-out test results, hybrid approaches worked consistently better than either pure content-based or pure collaborative approaches but no significant improvement was observed by exploiting high-degree association. Based on the results from pilot subject test, content-based approach outperformed collaborative and hybrid approach but no significant benefit of exploiting high-degree association was observed.

Extending the pilot subject test to a complete user study with real book selling experts will be part of our future work in this research. Other aspects of future works include fine-tuning the weighing schemes in the graph formation stage, and altering the parameter values in the Hopfield net algorithm. In addition, other graph search algorithms or recommendation methods can be applied to compare with our model.

In this study, we have tested the idea of using recommender systems for digital libraries in the context of online bookstore. We expect that our system can be applied in multilingual digital libraries to generate book recommendations based on book content and users' usage history. The evaluation of these applications will further test the usefulness of our model and system.

8. ACKNOWLEDGMENTS

This research is partly supported by NSF Digital Library Initiative-2, "High-performance Digital Library Systems: From Information Retrieval to Knowledge Management", IIS-9817473, April 1999-March 2002.

We would like to thank books.com.tw for providing the data set and their cooperation during the project. We would also like to thank Hui Liu for her involvement in system design and development.

9. REFERENCES

[1] Ahmad, M., Wasfi, A., Collecting User Access Patterns for Building User Profiles and collaborative Filtering. in *Proceedings of the 1999 International Conference on Intelligent User Interfaces*, (1999), 57-64.

[2] Baeza-Yates, R., Gonnet, G. Fast Text Searching for Regular Expressions or Automaton Searching on Tries. *Journal of the ACM*, *43* (6), (1996), 915-936.

[3] Balabanovic, M., Shoham, Y. Content-based, collaborative recommendation. *Communications of the ACM*, *40* (3), (1997), 66-72.

[4] Basu, C., Hirsh, H. Cohen, W., Nevill-Manning, C. Technical Paper Recommendation: A Study in Combining Multiple Information Sources. *Journal of Artificial Intelligence Research*, (2001). 231-252.

[5] Basu, C., Hirsh, H., Cohen, W., Recommendation as classification: Using social and content-based information in recommendation. in *Proceeding of the AAAI-98*, (Madison, WI, 1998), AAAI Press, 714-720.

[6] Chen, H., Ng, T. An Algorithmic Approach to Concept Exploration in a Large Knowledge Network (Automatic Thesaurus Consultation): Symbolic Branch-and-Bound Search vs. Connectionist Hopfield Net Activation. *Journal of the American Society for Information Science*, *46* (5), (1995). 348-369.

[7] Church, K., A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. in *Proceedings of the Second Annual Conference on Applied Natural Language Parsing ACL*, (Austin, TX, 1988).

[8] Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M., Combining Content-Based and Collaborative Filters in an Online Newspaper. in *Proceedings of ACM SIGIR Workshop on Recommender Systems*, (1999).

[9] Condliff, M.K., Lewis, D., Madigan, D., Posse, Bayesian, C., Mixed-effects Models for Recommender Systems. in *Proceedings of ACM SIGIR Workshop on Recommender Systems*, (1999).

[10] Dalton, J., Deshmane, A. Artificial neural networks. *IEEE Potentials*, *10* (2), (1991). 33-36.

[11] Geisler, G., McArthur, D., Giersch, S., Developing recommendation services for a digital library with uncertain and changing data. in *Proceedings of the first ACM/IEEE-CS Joint Conference on Digital libraries*, (Roanoke, VA, United States, 2001), 199 - 200.

[12] Hill, W., Stead, L., Rosenstein, M., Furnas, G., Recommending and evaluating choices in a virtual community of use. in *Proceedings of the Computer-Human Interaction Conference*, (Denver, CO, 1995), ACM Press, 194-201.

[13] Houston, A.L., Chen, H., Schatz, B.R., Hubbard, S.M., Sewell, R., Ng, T. Exploring the use of concept spaces to improve medical information retrieval. *Decision Support Systems*, *30* (2), (2000). 171-186.

[14] Knight, K. Connectionist ideas and algorithms. *Communications of the ACM*, *33* (11), (1990). 59-74.

[15] Kwok, K., Comparing Representations in Chinese Information Retrieval. in *Proceedings of ACM SIGIR*, (1997), 34-41.

[16] Manber, U., Myers, G. Suffix arrays: a new method for on-line string searches. *SIAM-Journal-on-Computing*, *22* (5), (1993). 935-948.

[17] Mooney, R., Roy, L., Content-based book recommending using learning for text categorization. in *Proceedings of the Fifth ACM Conference on Digital Libraries*, (2000), 195-204.

[18] Ong, T., Chen, H., Updateable PAT-Tree approach to Chinese key phrase extraction using mutual information: a linguistic foundation for knowledge management. in *Proceedings of the Second Asian Digital Library Conference*, (Taipei, Taiwan, 1999), 63-84.

[19] Pazzani, M. A Framework for Collaborative, ContentBased and Demographic Filtering. *Artificial Intelligence Review*, (1999), 393-408.

[20] Resnick, P., Varian, H. Recommender Systems. *Communications of the ACM*, *40* (3), (1997). 56-58.

[21] Salton, G. *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*. Addison Wesley, Reading, MA, 1989.

[22] Sarwar, B., Konstan, J., Borchers, A., Herlocker, J., Miller, B., Riedl, J., Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. in *Proceedings of the the ACM Conference on computer Supported Cooperative Work (CSCW)*, (1998).

[23] Schafer, J., Konstan, J., Riedl, J. E-Commerce Recommendation Applications. *Data Mining and Knowledge Discovery*, *5* (1-2), (2001). 115-153.

[24] Shardanand, U., Maes, P., Social Information Filtering: Algorithms for Automating 'Word of Mouth'. in *Proceedings of the Computer-Human Interaction Conference*, (Denver, CO, 1995), ACM Press, 210-217.