A Graph Model for E-Commerce Recommender Systems

A Graph Model for E-Commerce Recommender Systems

Zan Huang, Wingyan Chung, and Hsinchun Chen
Artificial Intelligence Lab, The Department of Management Information Systems, The University of Arizona, Rm. 430, McClelland Hall, 1130 East Helen Street, Tucson, AZ 85721. E-mail: zhuangQeller.arizona.edu

Information overload on the Web has created enormous challenges to customers selecting products for online purchases and to online businesses attempting to identify customers' preferences efficiently. Various recommender systems employing different data representations and recommendation methods are currently used to address these challenges. In this research, we developed a graph model that provides a generic data representation and can support different recommendation methods. To demonstrate its usefulness and flexibility, we developed three recommendation methods: direct retrieval, association mining, and high-degree association retrieval. We used a data set from an online bookstore as our research test-bed. Evaluation results showed that combining product content information and historical customer transaction information achieved more accurate predictions and relevant recommendations than using only collaborative information. However, comparisons among different methods showed that high-degree association retrieval did not perform significantly better than the association mining method or the direct retrieval method in our test-bed.

Introduction

The large amount of product information on the Web presents great challenges to both customers and online businesses in the e-commerce environment. Customers frequently experience difficulty in searching for products on the Web, while online businesses are often overwhelmed by the rich data they have collected and find it difficult to promote products appropriate to specific customers. Overload of product and transaction information on the Web raises a practical question: How can the large amount of product and usage information available from online transactions be effectively utilized to support better decision-making by both buyers and sellers?

To deal with this information overload problem, researchers have proposed recommender systems that automatically analyze users' usage data to filter Web page content, categorize newsgroup messages, and recommend good information resources (Resnick & Varian, 1997). Typically, a recommender system analyzes data about items' or about interactions between users and items to find associations among items and users. The results obtained are presented as recommendations. Using this concept, some online businesses, such as Amazon.com and eBay.com, have used recommender systems as a business tool. Such systems are reported to have enhanced e-commerce sales by converting browsers to buyers, increasing cross-selling and building customer loyalty (Schafer, Konstan, & Riedl, 2001). In this article, we review previous research in recommender systems to identify frequently used approaches and representations. Four recommendation approaches were examined: knowledge engineering, collaborative filtering, a content-based approach, and a hybrid approach. Different recommendation approaches can be implemented using different analytical methods. Commonly used methods are neighborhood formation, association rule mining, machine learning techniques, etc.

Recommender system developers often face two challenges. One is how to represent diverse information about the users and items. The other is how to build a model that is flexible enough to incorporate different recommendation approaches. We propose a graph model to represent the user-product information. The graph model contains nodes (customers and products) and links (transactions and similarities) that capture various types of e-commerce information. The model also supports different recommendation approaches. To demonstrate its flexibility, we have developed three analysis methods that cover a wide range of commonly used and newly introduced recommendation methods: direct retrieval, association mining, and high-degree association retrieval. With three methods all developed using a common model, we were able to obtain direct comparisons of performances of different recommendation methods under the three approaches of the collaborative

Received February 19, 2003; revised August 13, 2003; accepted August 13, 2003

 Published online 14 November 2003 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/asi.10372

[1] **"Items" refers to products, Web pages, persons with expertise, useful articles, etc.**

filtering, content-based, and hybrid approaches. We used a Chinese online bookstore as our test-bed to implement a recommender system based on the model and methods we developed. A systematic empirical evaluation was conducted and results were summarized. Initial results on the graph model and the high-degree association retrieval methods have been reported in a recent conference publication (Huang, Chung, Ong, & Chen, 2002).

Literature Review

Recommender systems assist users in choosing appropriate information items or products from a large set of alternatives. In such systems, personalized recommendations on items (information, products, or services) are generated by predicting preferences of users. While "recommender system" is viewed as a synonym of "retrieval system" by some researchers (Fumer, 2002), we argue that recommender systems are distinguished from typical retrieval systems by emphasizing the modeling of the user characteristics and user-item interactions in the system. In recommender systems, latent user preferences are assumed to be indicated by a wide range of observable data: features of the user, features of the items purchased by the user, behavior of users with (estimated) similar preferences, etc. The goal of recommender systems is to estimate such latent user preferences and provide predictions of appropriate items to specific users. Traditional retrieval systems typically focused on providing relevant information items that correspond to the explicit form of user preferences (i.e., queries) and did not emphasize the modeling of latent user preferences. Recommender systems can be implemented to enhance traditional retrieval systems, while also covering a wide range of new application areas, especially in the e-commerce environment.

Since the first recommender system was developed a decade ago (Goldberg, Nichols, Oki, & Terry, 1992), various techniques have been employed in various application-specific recommender systems. Early recommender systems helped users to sift through a large number of documents like Usenet news articles or Web pages. With the recent proliferation of e-commerce, recommender systems have become a powerful business tool to enhance customers' capability to overcome the product information overload problem. In this section, we summarize the literature and propose a taxonomy of recommender systems that characterizes a typical recommender system in three major dimensions: system input, data representation, and recommendation approach.

The input of a recommender system can take three forms: (1) user factual data such as name, gender, and address; (2) transactional data captured through users' explicit ratings or implicit feedback observed from their behaviors; and (3) item factual data such as product brand and price. Many recommender systems have employed users' explicit feedback in the form of ratings (Balabanovic & Shoham, 1997; Basu, Hersh, & Cohen, 1998; Claypool et al., 1999; Condliff, Lewis, Madigan, & Posse, 1999; Mooney & Roy, 2000; Passani & Billsus, 1997; Sarwar et al., 1998; Terveen, Hill, Amento, Mcdonald, & Creter, 1997). For example, *Syskills* and Webert used user ratings of a Web page as input data (Passani & Billsus, 1997). Basu et al. (1998) employed user ratings on movies as input to their system. This method suffers from the cold-start problem, in which newly released products may get only a few ratings and hence not be recommended even if they are relevant. To overcome this problem, some systems also have used implicit feedback obtained during the user's usage session. The system analyzes system logs to find users' preferences.

Existing recommender systems use different kinds of data representations that usually capture three basic elements in a recommender system—user representation, item representation, and transaction representation. User information can be represented in four ways: (1) by user attributes (usually demographic data such as gender, birth date, salary, etc.), (2) by associated items (e.g., the products the user has expressed interest in, has given ratings to or actually purchased), (3) by transactions (attributes extracted from the user's transaction history such as time, frequency, and amount, can partially represent a user's behavior pattern), or (4) by item attributes (e.g., a user may be characterized as liking romantic stories and favoring low prices based on the attributes of the books she has purchased). Most recommender systems represent users as a set of associated items; Basu et al., 1998; Claypool et al., 1999; Condliff et al., Fu, Budzick, & Hammond, 2000; Lieberman, 1995; Mladenic, 1996; Pazzani, 1999; Passani & Billsus, 1997; Sarwar et al., 1998; Shwab, Phol, & Koychev, 2000). For example, systems like Fab (Balabanovic & Shoham, 1997), Personalized Tango (Claypool et al., 1999), SurfLen (Fu et al., 2000), and Personalized Webwatcher (Mladenic, 1996) used Web pages the user had rated or previously visited to represent the user, whereas systems like GroupLens (Sarwar et al., 1998) used movies and books to represent the user interests. As another important element in recommender systems, items are usually represented by item attributes (such as price, content, brand, etc.) or by associated users (e.g., the customers who have purchased this item before). Some recommender systems also use transactions as a basic element, usually representing them by the transaction attributes (such as time, amount, etc.) or by items in the transactions. Researchers have also included some transaction attributes such as time and place as additional dimensions and support a different type of recommendation that may be based on different combinations of dimensions, such as the recommendation of Web content to a particular customer on weekends, or the recommendation of the best time to promote certain products to a particular customer (Adomavicius & Tuzhilin, 2001). However, most existing recommender systems focus on the analysis of the two dimensions of users and items.

Four approaches were used by recommender systems to generate recommendations, namely knowledge engineering,

collaborative filtering, content-based, and hybrid approaches.

The knowledge engineering approach seeks to discover the factors that affect users' preferences in choosing products. Significant human effort is needed to make this approach work. For example, Expertise Recommender (McDonald & Ackerman, 2000) recommends software engineering expertise to programmers using heuristics based on version change history and technical support records. The heuristics were created using the results obtained from a 5-month ethnographic study. Message filtering systems and document management systems have been using such a rule-based approach extensively. ISCREEN (Polloc, 1988), an early work in this area, used user-defined rules to screen text messages. The rules were personalized for each user, and the rules were mainly based on content and attributes of text messages (such as sender, length, highlights, etc.). The knowledge engineering approach can generate user-specific rules for recommendation purposes based on a wide range of information including item features, user features, and other domain-specific information. The major difference between this approach and the following approaches is that the process of generating such rules is not automated and extensive human effort is typically required.

The collaborative filtering approach obtains general collaborative information from users to form aggregate rules or trends as well as associations among users and/or items to predict the interest of individual users. An example is the movie recommendation, in which customer ratings on movies were used to recommend films to specific users. Most collaborative recommender systems are either user-based (Burke, 2000; Claypool et al., 1999; Mobasher et al., 2000; Nasraoui, Frigui, Joshi, & Krishnapuram, 1999; Pazzani & Billsus, 1997; Sarwar et al., 1998) or item-based (Sarwar, Karypis, Konstan, & Riedl, 2001). User-based collaboration uses historical data to form user neighborhoods. For example, Pazzani (1999) formed a neighborhood of similar users based on their expressed preferences and recommended restaurants based on the neighborhood relationship. Item-based collaborative recommendation relies on associations among items to determine recommendations. These associations are based on historical interactions between items and users, such as copurchase or preference by the same customer. Existing e-commerce recommender systems usually use this technique in applications such as gift recommendation (Schafer et al., 2001). Other collaborative recommender systems have used the user-item pair as the basis of the analysis, rather than just one of them. In some systems where the identity of the user is not available, collaborative filtering also can be applied on the basis of transactions. Several analysis techniques, including neighborhood formation, association rule mining, and machine learning techniques, typically have been applied to generate recommendations. Neighborhoods were formed by using similarity functions such as cosine-based similarity and correlation-based similarity functions and clustering techniques (Mobasher et d., 2000; Nasraoui et al. 1999; Sarwar et al., 2001). Association rule mining was used to extract patterns of association among items and users from the transaction history (Fu et al., 2000). Machine learning techniques were also applied to discover patterns from transactional data (Basu et al., 1998; Pazzani & Billsus, 1997). However, collaborative filtering does not work well when a product is newly introduced or a user just starts to use the system, because the system does not have much rating information on either the product or the user (the "cold-start problem"). Also, when there are many items but relatively few users, collaborative filtering cannot provide a good recommendation simply because there are few ratings (the "sparsity problem"). Furthermore, some users with opinions consistently different from the group opinions do not benefit from collaborative filtering (the "gray sheep problem") (Claypool et al., 1999).

The content-based approach borrows techniques from information retrieval to analyze the content of items. Neighborhood functions (Claypool et al., 1999; Liberman, 1995; Mladenic, 1996) and classification techniques (Mooney & Roy, 2000; Mostafa, Mukhopadhyay, Lan, & Palakal, 1997) often have been applied to analyze and cluster the textual content of items and recommendations are generated based on matching between item characteristics and user profiles. Although some systems have explored very comprehensive item representations [for example, Watters and Wang (2000) proposed a news object representation that consisted of a rich set of header, content, and behavior attributes for online news article recommendation], the content-based approach has several fundamental limitations. It captures only partial information on item characteristics, usually textual information. Other content information such as audio or visual content is usually ignored. This approach tends to recommend only items with similar characteristics (also known as the "over-specificationproblem"). Only the target user's feedback is used in this approach, although that user's interest may also be influenced by other users' preferences (Balabanovic & Shoham, 1997).

To improve the quality of recommendation, many efforts have been made to combine content-based and collaborative approaches to implement a hybrid approach. Recommender systems using a hybrid approach can be loosely categorized into three classes. Systems in the first category try to integrate the two approaches by simply merging the two recommendation results separately generated by content-based and collaborative approaches (Claypool et al., 1999). Systems in the second category combine the two approaches at the representation level, in which product information and transaction information are incorporated into a single representation. Some of these systems have been based on collaborative filtering and added content information of items into the user representation or added ratings based on content of the items (Pazzani, 1999; Sarwar et al., 1998; Singh, Yu, & Venkatraman, 2001). For example, *Fab* maintained user profiles based on content analysis of documents rated by users. It then compared user profiles to identify similar users to generate a collaborative recommendation

(Balabanovic & Shoham, 1997). Others tried to apply content-based analysis and added user information into the item representation (Goldberg et al., 1992). Some other systems have been built on a comprehensive model that incorporated different information sources (Basu et al., 1998; Condliff et al., 1999). Recent work by Ansari, Essegaier, and Kohli (2000) has adopted the formal statistical approach to model the user ratings as a function of item attributes, user characteristics, and expert evaluations. Unobserved sources of heterogeneity in user preference and item appeal structures were accounted for using this approach. Systems in the third category of hybrid approach use knowledge engineering in addition to collaborative filtering. For example, Burke proposed a hybrid recommendation framework in which collaborative filtering was used to postfilter the recommendation results obtained by the knowledge-based filtering method (Burke, 2000).

Table 1 presents the taxonomy of recommender systems described above. We summarize in the table the types of system input, representation methods, and recommendation approaches used in existing recommender systems. Several systems in the literature are listed in Table 2 and are characterized in Table 3 under the dimensions introduced in the taxonomy. We found that most of the recommender systems used the collaborative filtering approach to generate recommendations. Also, the methods used in these systems were largely application-specific. This limited the data inputs and representations that could be used. We believe that a model not only needs to be comprehensive enough to support diverse inputs and representations but also flexible enough to support different recommendation approaches.

A Graph Model for E-Commerce Recommendation

To address the needs of comprehensive representation and to support flexible recommendation approaches, we propose a graph model for building e-commerce recommender systems.

Two recent works on recommender systems propose graph-theoretic models or approaches. Aggarwal, Wolf, Wu, and Yu (1999) introduced a directed graph of users in recommender systems, where the directed edges correspond to the notion of predictability. Based on this graph, personalized recommendations can be generated via a few reasonably short (strongly predictable) directed paths joining multiple users. Mirza (2001) also proposed a graph-theoretic model for collaborative filtering, in which items and users are both represented as nodes and the edges represent the recommendation data set (interaction between user and items). A social network graph of users is then created based on the original graph, and recommendations are generated by navigating the combination of the original graph and user social network graph. While similar to these previous graph-theoretic models in some aspects, the model we are proposing focuses more on providing a unifying representation for the various types of system inputs summarized in the taxonomy presented previously. Flexible recommendation approaches are also to be explored using such a graph model.

In the following section, we first describe the model and then analyze its properties and application. As we are focusing on e-commerce recommender systems, we will discuss products (instead of items) and customers (instead of users) in the following sections.

A Two-Layer Graph Model

Figure 1 shows a two-layer graph model, in which the two layers of nodes represent products and customers, respectively. Three types of links between nodes capture the input information: the product information, customer information, and transaction information. Each link between two products captures similarity between them. Different types of product information can be used to compute the similarity. For products like books and movies, content of the product description (book introductions, movie reviews) can also be used to compute product similarity.

Similarly, each link between two customers captures the similarity between them. The large amount of customer demographic data available at e-commerce sites can be used to compute customer similarity. Other information about customers may also be added, such as customers' answers to questionnaires, query inputs, Web usage patterns and so on.

Interlayer links are formed based on the transaction information that captures the associations between customers and products. Some commonly used transaction information includes purchase history, customers' rating, or browsing behavior that involves the product. Each purchase of a product by a customer is represented by an interlayer link in our model. Different types of transaction information may be combined in the model by assigning different weights to reflect different association strengths. For example, a high rating on a product may be weighted higher than a browsing activity, because the former reflects the customer's interest more directly.

Graph Model Analysis

The literature review showed that the field of recommender systems calls for a comprehensive and generic model. This model should have the capability to represent different types of data inputs and to support different recommendation ,approaches using various techniques. Here we show how our two-layer graph model meets these requirements. We first analyze the properties of our model and introduce three recommendation methods we have developed to show the usefulness and flexibility of the model.

Data representation. The major difference between the representation in our model and other representations is that our model requires transformation of the original input data, such as product and customer attributes, into similarity measures. Others have used the original input data itself as

TABLE 1. A taxonomy of recommender systems.

System input		
Type		Details
User factual data		Contains the user's demographic information such as name, social security number, gender, birth date, salary, telephone number, and address.
Transactional data	Explicit feedback	Consists of records of the users' explicit expression of the interest on the item, such as rating and comments. Multi-scale ratings are also used.
	Implicit feedback	Contains the interaction between users and items, including examination (selection, duration, edit ware, repetition, purchase), retention (save, annotation, organization, print, delete), and reference.
tem factual data		Consists of the attributes of an item such as product brand, price, Web page content, and links to other pages.

Data representation		
Type and Method		Details
User	By user attributes	Represents a user by the attributes (such as gender, birth date, salary) he/she has.
	By items associated	Represents a user by the items he/she has interacted with (e.g., products purchased by a customer).
	By transactions	Represents a user by what he/she has done (e.g., the past purchase transaction).
	By item attributes	Represents a user by attributes of the items that he/she has interacted with (e.g., use important keywords in the Web pages the users viewed before to present his/her interests).
	By item attributes	Represents an item by its attributes such as its price, color, weight, brand, and so on.
	By users associated	Represents an item by the set of users who have bought it.
Transaction	By transaction attributes	Represents a transaction by the attributes it has (e.g., day of purchase, time of purchase, sequence of shopping).
	By items	Represents a transaction by the items in it.

Recommendation approach				
Type and Method			Details	Problems
Knowledge engineering			Uses human efforts or heuristics to identify factors that determine users' interests and generate recommendations (case-based reasoning, decision support tools).	Significant amount of time is involved. Highly application specific and strong dependence on the availability of persons with expertise.
Collaborative filtering	Basis	User-based	Uses historical transactional data to form user neighborhoods.	Early users may not get good recommendations because of initial sparse ratings; users who hold different opinions with the community may not get good recommendations; the input data usually contains large number of users nad items and is high dimensional, which makes the system difficult to scale up; recommendation quality reacts slowly to drift; need incentive for users to give recommendations (Claypool et al., 1999; Sarwar et al., 1998).
		Item-based	Uses historical transactional data to form item neighborhoods.	
		User and item-based	Uses pair of user and item, not based on just one of them.	
		Transaction-based	Uses transaction attributes to form transaction neighborhoods.	
	Technique	Neighborhood formation	Generates neighborhoods of users or items using different methods, including correlation (Pearson), similarity functions (Cosine), clustering techniques (PACT, Clique, Hypergraph).	
		Association rule mining	Extracts patterns of association among items or users using association rule mining.	
		Machine learning	Inductively learns patterns of association within a given set of training data. Algorithms include Bayesian network and neural network.	
Content-based		Neighborhood formation	Uses co-occurrence analysis or similarity functions to form .neighborhood.	Only textual content information can be captured, but not audio or visual content; only similar items can be recommended; only the target user's feedback can be used (Balabanovic & Shoham, 1997).
		Classification	Uses machine learning algorithms (text categorization), including Bayesian classifier, Winnow, Rocchio's algorithm.	
Hybrid	Collaborative filtering + content-based	Merging results of different approaches	Uses ad hoc weights to combine content-based and collaborative filtering results.	Many systems are tailored to some specific applications and hence, not reusable. Some may need heuristics and human engineering effort that requires significant amount of time.
		Collaborative filtering augmented by content information	Uses collabortive filtering framework, and adds content information of the terms that users interacted with to user representation	
		Content-based approach augmented by collaborative information	Applies content-based analysis on items selected by user5s and then identified items that similar users would choose.	
		Comprehensive model	Builds comprehensive model that incorporates diffemt information sources.	
	Collaborative filtering + knowledge engineering		Integrates knowledge-based system with collaborate filter system.	

TABLE 2. Examples of recommender system research.

Research/System	Description
	It screens text messages based on user-defined rules. The rules were constructed mainly based on the content and properties of the messages. A conflict detection component was also included to examine rules for consistency.
Goldberg et al., 1992; Tapestry	In addition to content-based filtering, it supports collaborative filtering to allow users of an e-mail system to annotate documents that are interesting (or uninteresting) to them.
Lieberman, 1995; Letizia	It recommends interesting items on the Web that are related to a user's current navigation context based on a set of links and user profiles which consists of a list of weighted keywords. It uses heuristics to determine positive and negative evidence of a user's information interest.
Mladenic et al., 1996; Personalized Webwatcher	It watches users' choices of URL links on Web pages to recommend links on other Web pages that are visited later. Visited URL links are taken as positive examples and nonvisited links as negative examples.
Balabanovic, 1997; Fab	It suggests relevant URLs to users by combining users' ratings and Web pages' similarities.
Mostafa et al., 1997; SIFTER	It achieved intelligent information filtering by including a vector-space model for document representation, an unsupervised learning component for document classification, and a reinforcement learning component for user modeling.
Pazzani et al., 1997; Syskills & Webert	It classifies and suggests Web pages that a user may be interested in. Based on Web pages with users' ratings, a set of word-probabilities triplets is then formed for each user.
Terveen et al., 1997; PHOAKS	It employs complicated categorization rules to identify recommended web resources in Usenet messages. These recommended web resources are ranked by distinct recommenders and were presented as recommendation results. In this system, no personalization is implemented, where each user gets the same recommendations.
Basu et al., 1998	It uses collaborative features derived from users' rating on movies as well as the content feature of the movies to form the feature vector of a user-movie pair, and uses inductive learning methods to predict like or dislike for each user and movie combination.
Sarwar, 1998; GroupLens	It provides a collaborative filtering solution for Usenet news and movies. Heuristics such as spellchecking, attached-message checking, and message-length checking are embedded in content analysis.
Burke, 1999; Entree	It recommends restaurants that meet the user's interests and budget. The system integrated knowledge-based and collaborative-filteringrecommender systems by using semantic ratings in which a system obtains reasons behind a preference.
Claypool et al., 1999; Personalized Tango	It provides personalized filtering for an online newspaper. The basis for content-based and collaborative predictions were kept separate. This allows for individual advances made to either the content-based or collaborative filter.
Condliff et al., 1999	Using a Bayesian methodology, it recommends drinks/movies based on users' ratings, user features, and item features.
Nasraoui et al., 1999	It uses clustering techniques to analyze server access logs and obtain typical session profiles of users to recommend Web pages.
Pazzani, 1999	It recommends restaurants based on users' preferences as described by Web pages.
et al., 2000; SurfLen	It recommends Web pages based on navigation history.
McDonald & Ackerman, 2000; Expertise Recommender	It locates expertise to solve problems in an organization. A range of recommendation problems that involved some expertise and persons acts as input to the system.
Mobasher et al., 2000	It creates aggregated user profiles based on Web usage data and transactions to recommend Web pages.
Mooney & Roy, 2000	It is a content-based book recommender system that utilizes information extraction and a machine-learning algorithm for text categorization.
Ansari et al., 2000	It uses a hierarchical Bayesian approach to model customer ratings as a function of product attributes, customer characteristics, and expert evaluations. Unobserved sources of heterogeneity in customer preferences and product appeal structures are accounted for in their model.
Schwab et al., 2000; Electronic Funding Information (ELFI)	It provides suggestions on what funding programs and agencies are suitable for a researcher. Interaction log files are used to create a user profile that contains information about the detailed views (DV) selected by the user. Bayes' Theorem and k-Nearest Neighbor were used to cluster users' interests; TF/IDF was used in analyzing 189 most discriminating words in DV.
Adomavicius & Tuzhilin, 2001; 1:1Pro	It constructs personal profiles based on customers' transactional histories. The system uses data mining techniques to discover a set of rules describing customers' behavior.
Singh et al., 2001	It enables users to locate desirable services based on trustworthy, personalized recommendations of their peers. The information comes from the social network regarding the reputation and sociability of participants. It uses a unique way of collaborative filtering where recommendations are generated based on a user-customized social network.
Sarwar et al.. 2001	It analyzes different item-based collaborative filtering recommendation generation algorithms. Different item similarities and techniques for obtaining recommendations were explored. It was found that item-based collaborative filtering achieves better performance and higher quality than best available user-based algorithms.
Lin et al., 2002	It performs collaborative recommendation based on a new association rule algorithm. Rules between users and rules between items are used to generate recommendations. Experimental results showed that their algorithm achieved significantly better performance than that of traditional correlation-based approaches.
Chau et al., 2002	It stores user's Web page search sessions, which includes users' feedback on the keywords and returned URLs in the session as well as user comments. The related search sessions are given as collaborative recommendations based on the session attributes, which are search topics chosen by the user.

TABLE 3. **System inputs, data representation, and approaches used in example recommender system research.**

Research/System	System input	Data representation	Approach
Pollock, 1988; ISCREEN	Item factual data		Knowledge engineering
Goldberg et al., 1992; Tapestry	Transactional data—implicit feedback, item factual data	Item—by users associated	Hybrid–collaborative filtering + content-based-content-based approach augmented by collaborative information
Lieberman, 1995; Letizia	Transactional data—implicit feedback, item factual data	User-by items associated	Knowledge engineering, content-based–neighborhood formation
Mladenic et al., 1996; Personalized Webwatcher	Transactional data—implicit feedback	User—by items associated	Content-based–neighborhood formation
Balabanovic, 1997; Fab	Transactional data—explicit feedback	User-by items associated	Hybrid-collaborative filtering + content-based–collaborative filtering augmented by content information
Mostafa et al., 1997; SIFTER	Item factual data, transactional data-explicit feedback	Item—by item attributes, use—by item attributes (relevance probability vector of classes of documents)	Content-based-classification
Pazzani et al., 1997; Syskills & Webert	Transactional data—implicit feedback, transactional data-explicit feedback	User—by items associated	Content-based-classification
Terveen et al., 1997; PHOAKS	Transactional data—explicit feedback	Item—by users associated	
Basu et al., 1998	Transactional data-explicit feedback, item factual data	User—by items associated, item-by item attributes, and users associated	Collaborative filtering—basis–user and item-based, collaborative filter—technique–machinelearning, content-based-classification, hybrid–collaborativfe filtering + content-based–comprehensive model
Sarwar, 1998; GroupLens	Transactional data-explicit feedback	User—by items associated and transactions, item—by users associated	Collaborative filtering—basis–user-based,collaborative filtering—technique–neighborhood formation, hybrid–collaborative filtering + content-based–content-based approach augmented by collaborative information
Burke, 1999; Entrée	Transactional data—implicit feedback	User—by transactions, item–by users associated	Knowledge engineering, collaborative filtering—basis–user-based. hybrid–collaborative filtering + knowledge engineering
Claypool et al., 1999; Personalized Tango	User factual data, transactional data-explicit feedback	User—by items associated, and items attributes	Collaborative filtering—basis–user-based,collaborative filtering—technique–neighborhood formation. content-based–neighborhood formation, hybrid–collaborative filtering + content-based–merging results of different approaches
Condliff et al.. 1999	User factual data, transactional data-explicit feedback, item factual data	User—by user attributes, and items associated, item—by item attributes, and users associated	Collaborative filtering—basis-user and item-based, hybrid–collaborative filtering + content-based-comprehensive model
Nasraoui et al., 1999	Transactional data—implicit feedback	User—by transactions, transaction—by items	Collaborative filtering-basis-user- bas^, collaborative filtering—technique–neighborhood formation
Pazzani, 1999	User factual data, transactional data-explicit feedback, item factual data	User—by user attributes, items associated, and item attributes, item—by item attributes	Collaborative filtering—basis–user-based,collaborative filtering—technique–neighborhood formation, content-based–neighborhood formation, hybrid-collaborative filtering + content-based-collaborative filtering augmented by content information
Fu et al., 2000 SurfLen	Transactional data—implicit feedback	User—by items associated, item—by users associated	Collaborative filtering—basis–user and item-based, collaborative filtering—technique–associationrule mining
McDonald & Ackerman, 2000; Expertise Recommender	Transactional data—iimplicit feedback, item factual data	Item–by item attributes	Knowledge engineering
Mobasher et al., 2000	Transactional data—implicit feedback	User—by transactions, transaction—by items	Collaborative filtering—basis–user-based,collaborative filtering technique–neighborhood formation
Mooney & Roy, 2000	User factual data, transactional data-explicit feedback	Item–by item attributes	Content-based–classification
Ansari et al., 2000	User factual data, item factual data, transactional data-explicit feedback	User—by user attributes, item—by item attributes, transaction—by items	Hybrid–comprehensive model
Schwab et al., 2000; Electronic Funding Information (ELFI)	Transactional data—implicit feedback	User—by items associated, and transactions	Knowledge engineering, content-based-classification
Adomavicius & Tuzhilin, 2001; 1: 1Pro	User factual data, transactional data—implicit feedback	User—by user attributes and transactions, transaction—by transaction attributes and items	Knowledge engineering
Singh et al., 2001	Transactional data—implicit feedback	User—by transactions	Hybrid–collaborative filtering + content-based-collaborative filtering augmented by content information
Sarwar et al., 2001	Transactional data-explicit feedback	Item–by user associated	Collaborative filtering—basis–item-based
Lin et al., 2002	Transactional data-explicit feedback	Item—by users associated, user—by item associated	Collaborative filtering—basis–item-based and user-based, collaborative filtering—technique—association rule mining
Chau et al.. 2002	Transactional data-explicit feedback	Transaction-by transaction attributes	Collaborative filtering—basis–transaction-based

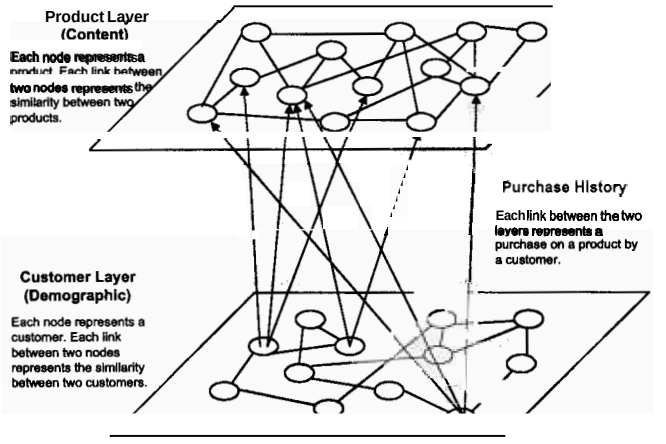

FIG. Two-layer graph model of products, customers, and transactions.

representation. In general, we may lose some information by conducting this transformation because the only information about the customer or product is relative similarities to other products or customers. However, since different neighborhood formation techniques are usually applied in recommender systems, and recommendations are usually generated based on the results of neighborhood formation, the transformation we performed did not cause loss of any significant information. At the same time, the benefit of this transformed representation is that it makes the model compact and generic by using a graph to capture all related information.

Using this model, product and customer factual information is preserved in the interlayer similarities. By assigning different meanings to interlayer links, both explicit and implicit feedback information can be captured. This two-layer model captures all types of data inputs and covers most data representations that have been summarized in the Literature Review section.

Recommendation approach. In this research, we considered only content-based, collaborative filtering, and hybrid approaches that can be implemented by automatic techniques (but not knowledge engineering). By choosing different types of links to use in a recommendation generation process, content-based, collaborative, and hybrid approaches can be performed using our model. If only product information is used, which means only links in the product layer are activated, it is a content-based approach. If customer-layer and interlayer links are activated, it is a collaborative approach. If all links are activated, it becomes a hybrid approach. The model is flexible because different combinations of links can be activated at run time. For example, we can choose to use product description and price to form product similarity, use demographic data to form customer similarity, and use customers' ratings and purchase history to form interlayer links. The representation can even be personalized for each customer by using a customized weighting scheme. For example, if a customer relies more on the collaborative recommendation, we can assign higher weights to interlayer links and customer-layer links. If the customer is more price-driven, we can assign higher weights to the price information of the product when computing the similarity between products.

Recommendation Methods for the Graph Model

Different recommendation methods can be developed based on our model. By viewing the recommendation process as a graph search, we use our model to generate recommendations by finding nodes that have high associations with the starting nodes. Many recommendation methods can be developed based on this graph view. We developed three recommendation methods that covered a wide range of methods being used in the field. For each method, we also describe how collaborative, content-based, and hybrid recommendation approaches can be performed.

Direct retrieval. The direct retrieval method generates recommendations by retrieving products similar to the target customer's previous purchases and products purchased by customers similar to the target customer. It covers most of the recommendation methods that form neighborhoods of customers or products. For content-based recommendation, products that are similar to the target customer's previous purchases are retrieved as recommendations. This situation is similar to document retrieval, in which documents similar to the input queries are retrieved. For collaborative recommendation, a list of customers similar to the target customer is first obtained. Then, similarities can be traced through customer-layer links or through interlayer links from the target customer's previous purchases. In other words, similar customers are those who are demographically similar to the target customer or those who bought the same books as the target customer. The products that are linked to this set of similar customers are retrieved as the collaborative recommendation for the target customer. The hybrid recommendation is obtained by combining the recommendation results from the two approaches described above.

Association mining. The association mining method is a data mining technique commonly used in many traditional and online retailing businesses. Recent research such Lin, Alvarez, and Ruiz, (2002) showed that association mining outperformed traditional recommendation methods. Having been used for collaborative filtering, this method uses transaction history as input and generates association rules about the purchase patterns to predict customers' purchases. Recent development of association mining algorithms provides better support for collaborative filtering. For example, new algorithms have been developed to explore the correlation between items (Aggarwal & Yu, 2001; Cohen et al., 2001); and to specifically support collaborative filtering (Lin et al., 2002). However, since association mining is not the main focus of this research, we chose to use the standard associ-

ation-mining algorithm to construct a benchmark recommendation method for comparison with the other two methods under our graph model. We generalized the standard association mining method for content-based, collaborative, and hybrid recommendation approaches by defining item sets in different ways.

The first step in association mining is to generate association rules for content-based recommendation, collaborative recommendation, and hybrid recommendation. An association rule is defined by its rule body and rule head, each containing a set of products.[2] It can be stated as ***Rule body*** $\Rightarrow$ ***Rule head,*** meaning that if a customer buys products in the ***Rule body*** (the condition), then he/she is likely to buy products in ***Rule head*** (the result). In the following examples, association rule 1 has one book on its left-hand side and one book on its right-hand side. This means that if a customer buys book **B1** then he/she is likely to buy book **B2.** Rule **2** tells us that if a customer buys book **B3** and book **B2** then he/she is likely to buy B1. From the lift values (which indicate the strength of association between the two sets of products), we know that rule **2** is stronger than rule **1.**

Examples of association rules:

Rule 1 (lift = 8.0600) B1 $\Rightarrow$ B2
Rule 2 (lift = 9.7600) B2+B3 $\Rightarrow$ B1

The following notations refer to books.

B1 = **"Harry Potter and the Chamber** of **Secrets"**
B2 = **"Harry Potter and the Philosopher's Stone"**
B3 = **"Harry Potter and the Prisoner** of **Azkaban"**

For content-based recommendation, our procedure for generating association rules is as follows. For each product, we retrieve a number of most similar products ranked by similarity weights. This number can vary depending on how much output the customer wants. These products form an association set that is used for association mining. Therefore, we have N sets of similar products for the entire product collection where N is equal to the total number of products. To find out which associations are significant, we use statistical measures in association data mining to define whether the patterns observed are significant. One important measure is called "support," which means the relative occurrence of the detected association rules within the entire database of purchase transactions. The smaller this number is, the more patterns we can identify as associations. These patterns are then treated as association rules. The association rules obtained in this operation are called content association rules.

For collaborative recommendation, we generate association rules by treating the purchase history of each customer as a transaction. The products in this transaction form an association set that **is** used for association mining. As a result, we obtain M sets of products where M equals the total number of customers who have previously purchased products. As we are interested in predicting which products are likely to be purchased by a customer, given his/her purchase history, we use the measures described above (support and lift values) to discover the association rules. These rules are called purchase association rules.

Once the association rules are generated, we can proceed to make recommendations based on different approaches. Our approach is generic because it can produce content recommendations, collaborative recommendations or hybrid recommendations, based on the type of association rules used. If content association rules are used, the approach is content-based. If purchase association rules are used, the approach is collaborative. If both content and purchase association rules are used, the approach is hybrid.

To make recommendations, our system first retrieves the customer's purchase history. Then, for each product in the purchase history, it applies all association rules that contain this product in their rule bodies to produce a set of products and to remove products that duplicate those in the purchase history. Finally, the system retrieves the top 50 (or fewer, if fewer than 50 rules are present) products from all the sets of products retrieved and ranks them in descending order of the aggregate lift values. If a product appears more than once, the lift values are summed to become an aggregate lift value.

High-degree association retrieval. Sparsity (the sparse rating problem) is a major problem in recommender systems (Sarwar et al., **2001).** The number of customers' ratings or purchases may be very small compared with the large number of products in the system, especially for commercial recommender systems with very large product sets (e.g., Amazon.com and CDnow). The recommender system may be unable to generate recommendations for many customers, which is a problem known as reduced coverage. Previous research has found that dimensionality reduction methods are promising to overcome this problem (Sarwar et al., **1998).** Based on our graph model, we propose another method for solving the sparsity problem. We increase the number of associations by exploring transitive neighborhoods rather than by reducing dimensions. We refer to this method as high-degree association retrieval and describe the details in this section.

A simplified example is used to formulate the problem of high-degree association recommendation in our graph model.

A recommendation is generated based on the association strength between a customer and a product. In the example depicted in Figure **2** the association between C1 and P1 is to be estimated. If only one-degree association is allowed, then no association between C1 and P1 can be inferred. In that case, association between customer and product exists only when a customer actually has bought the product. When

[2] **We use the terms and definitions appearing in "Association concepts," IBM DB2 Intelligent Miner for Data—Using the association visualizer, Version *6*, Release 1, 1st Edition, IBM Corporation,** 1999, **pp. 1–8.**

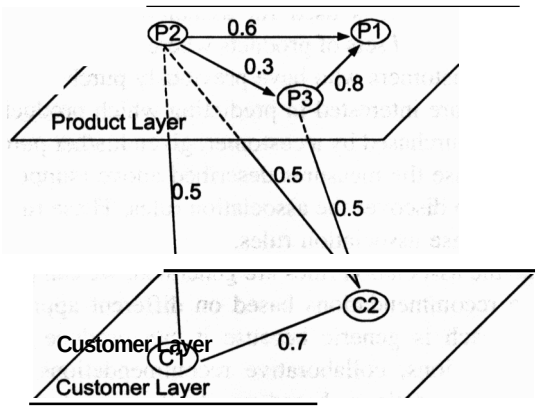

FIG. 2. Simplified two-layer graph example.

two-degree association is allowed, the path of **C1-P2-P1** can be utilized to form an association estimate between **C1** and **P1,** where the path consists of one purchase link **C1-P2** and one product similarity link **P2-P1.** 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 **P1** is $0.5 \times 0.6 = \mathbf{0.3,}$ and this result is actually based on a pure content-based approach. When three-degree association is allowed, there will be three more paths from **C1** to **P1, C1-C2-P2-P1, C1-P2-P3-P1,** and **C1-C2-P3-P1.** By summing up association measures of all of these four paths, we get the three-degree association measure of **C1** and **P1,** which is $\mathbf{0.3 + 0.21 + 0.12 + 0.28 = 0.91.}$ The choice of using the summation rather than average or maximum of the individual path association is based on the fact that the associations captured in individual paths contribute to the possible preferences of customers in an additive manner. For example, products that attracted more customers are more likely to also attract additional customers. Although a more delicate weighting scheme for the summation can be explored to improve the accuracy of this preference estimate, we think a simple summation captures the most important information and leave the weighted summation for future research. The result obtained from adding path associations is, in fact, a combination of the content-based and collaborative approaches. When high-degree association is allowed, it becomes computationally intensive to find out which nodes have strong associations with the target node.

High-degree association retrieval has been widely discussed in the information retrieval literature. "Spreading activation" has been used successfully for this purpose in various applications and several algorithms have been presented in the literature. Two representative algorithms are the branch-and-bound serial search algorithm and the neural net parallel relaxation algorithm (Chen & Ng, **1995).** Although the branch-and-bound algorithm required less computation time than the Hopfield net algorithm for the same retrieval task, the Hopfield net algorithm was able to produce more even activation of nodes from the network than branch-and-bound. In addition, the Hopfield net algorithm was reported to be suitable for related concept retrieval (Houston et al., 2000), which was similar to the highly associated products retrieval in our recommender system model. Therefore, we chose the Hopfield net as our spreading activation algorithm for producing high-degree association. We briefly describe the Hopfield net algorithm for high-degree association recommendation here, but other methods can also be adopted in our generic model.

The Hopfield net algorithm performs a parallel relaxation search during which nodes are activated in parallel and activation values from different sources are combined for each individual node. Neighboring nodes are traversed 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 or customers. This algorithm uses an iterative activation process:

$$\mu_j(t + \qquad \sum^{n-1} t_{ij}\mu_i(t) \qquad 0 \leq \quad \leq n -$$

where $\mu_j(t + 1)$ is the activation value of nodej at iteration $t + 1$, t_{ij} equals the association weight between node i to node j if there is a link that points from node i to node j, otherwise t_{ij} equals 0, and f_s is the SIGMOID transformation function (normalize any value to between 0 and 1). The algorithm terminates when there is no significant difference in terms of output between two iterations. The product nodes with highest weights in the final state of the network are retrieved as recommendations.

Similar to the direct retrieval method and association mining, high-degree association retrieval can support collaborative filtering, content-based and hybrid approaches by activating different types of links in the two-layer graph model.

Based on the two-layer graph model we have proposed, we believe that the two-layer graph model could capture different types of data inputs using a unified representation and could support different recommendation approaches. Based on the model, we also developed three recommendation methods for generating recommendations. The direct retrieval and association-mining methods cover a wide range of commonly used recommendation methods in the literature. We also developed a new high-degree association retrieval method to address the recommendation sparsity problem. We believe the two-layer model we propose could help unify past research and provide new insights.

Comparing Graph-Based Recommendation Approaches and Methods

We compared the performance of different graph-based recommendation approaches and methods that have been

proposed. We aimed to address two research questions. First, we wanted to find out whether combining content-based information and collaborative information (a hybrid approach) would improve the quality of recommendation. Second, we wanted to know whether there were significant differences among the three recommendation methods proposed. In particular, we wanted to see if the high-degree association retrieval method performed better than the direct retrieval and association mining method. The following section describes our research test-bed, implementation of the three recommendation methods, and an empirical evaluation study.

Research Test-Bed

We used transactions obtained from an online Chinese bookstore (one of the largest online bookstores in Taiwan) as our research test-bed. Covering the purchase and book information for **5** recent years, our data consisted of three types of information: books, demographics, and transactions. There were **9695** books, 2000 customers, and 18,771 transaction records in the data set. We preprocessed the data by extracting the Chinese key phrases from book content and computing similarity weights for books and customers.

Mutual information. A mutual information algorithm was used to summarize the co-occurrence of Chinese characters in a corpus to identify key phrase patterns. "Mutual information" is a metric that measures how frequently a character pattern appears in a Chinese corpus, relative to its subpattems. In its simplest form, the mutual information for a pattern ***c (MI,)*** is expressed as

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

where f stands for the frequency of a set of Chinese characters. Intuitively MI_c represents the probability of co-occurrence of the pattern c (e.g., 人工智慧 meaning artificial intelligence), relative to its left subpattern (人工智) and the right subpattern (工智慧). If MI_c is close to 1, pattern c is more likely to form a phrase than its left and right subpatterns alone. On the other hand, if MI, is close to 0, pattern c is not likely to form a phrase. An important algorithmic requirement for this approach is an efficient data structure that makes it feasible to analyze a large collection of training corpuses (Ong & Chen, **1999).** However, a greater challenge is finding all possible patterns that exist in the corpus along with their frequency of 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 to accessing a large corpus. In addition, previous research in text searching suggests that other similar data structures, including suffix array or PAT-array, could be equally efficient (Baeza-Yates & Gonnet, **1996;** Manber & Myers, **1993).**

The mutual information approach could be seen as an extension to bi-gram, tri-gram, and even n-gram in traditional information retrieval techniques. Church **(1988)** pointed out that some Bible literature has repeated patterns with up to 400 words, making it a challenge for the n-gram technique, because without removal, every subpattern in the 400-word sentence will be extracted. Removal of a pattern from the corpus affects the frequency distribution of the corpus, especially repetitive removals of many subpatterns. To solve this problem, we used a new data structure to support consistent online frequency update after removing patterns that we had already extracted (Ong & Chen, **1999).** This method increased the success of subsequent extractions, because the 400-word sentence now was extracted only once and its subpatterns were not extracted.

Co-occurrence analysis. We represented customers and books by feature vectors from factual customer and book information. The book feature vector was composed of two parts: textual content and book attributes. Textual content was represented by weighted Chinese phrases extracted from book title, introduction, foreword, author, and keyword. Book attributes included other aspects such as publisher, category, book layout, and so on. When computing the similarity between two books, we compared the content part and the attribute part of the two book feature vectors.

For the computation of the content similarity between two books, Salton's vector-space model (Salton, **1989)** was adopted. The combined weight of descriptor ***i*** in book j was computed by multiplying the ***term frequency*** (tf_{ij}) and ***inverse document frequency*** (log $[(N/df_i) \times w_i]$) as follows

$$d_{ij} = tf_{ij} \times \log[(N/df_i) \times w_i] \times pw, \qquad 1)$$

where w_i represents the number of Chinese characters in descriptor ***i*** and pw_i represents the weight of descriptor ***i.*** A descriptor can be a term or phrase containing any number of Chinese characters. The book feature vector was composed of textual content and book attributes. An asymmetric clustering function (Chen & Lynch, 1992), that has been shown to perform better than other clustering functions like cosine function in computing similarity for term associations, was used to compute the book content similarity. The asymmetric similarity from bookj (B_j) to book k (B_k) is given in Equation (2) while the asymmetric similarity from book k (B_k) to bookj (B_j) is given in Equation **(3).**

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

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

where d_{ij} is the combined weight of descriptor ***i*** in bookj as computed in Equation (1);

$d_{ijk} = tf_{ijk} \times \log [(N/df_{jk}) \times w_i]$ is the combined weight of descriptor i in the combination of books j and k, where N represents the total number of books;

tf_{ij} represents the number of occurrences of descriptor i in book j;

$tf_{ijk} = tf_{ij}$ when descriptor i appears in both book j and k, otherwise $tf_{ijk} = 0$;

df_j represents the number of descriptors that appear in book j;

tf_{ik} represents the number of common descriptors appearing in both book j and book k.

Book content similarity was asymmetric because the similarity weight from book j to book k might not be the same as the similarity weight from book k to bookj, based on their textual content. The book attribute similarity was made symmetric because two books can either have or not have the same value for those attributes. Content similarity and book attribute similarity were combined to form the similarity between two books.

Implementation of Recommendation Methods

With the book similarity weights and customer similarity weights available, we could generate recommendations using the three recommendation methods described above.

Direct retrieval. For the direct retrieval method, we developed several stored procedures in the database to generate recommendations following the method we described in the above section. The method is simple and can be implemented with simple SQL queries. For each approach, collaborative filtering, content-based and hybrid, we generated a recommendation list of 50 books for each customer. This method is vulnerable to the sparsity problem. When customers in the system did not make many purchases, using this direct retrieval method may not have been able to generate many recommendations.

Association mining. For the association mining method, we used the IBM Intelligent Miner as our data mining tool because it is popular in the industry and is available to our research lab. We followed the steps described in the previous section to generate association rules for content and collaborative recommendation. For content-based recommendation, since the number of pairs of similar books in our database was very large (855,727 pairs), we needed to determine a suitable number of the most similar books to be retrieved to produce a set of content association rules of manageable size. This number can be represented as a proportion of the total number of similar books. For a given book, we decided to consider only the top 50% (or fewer) of books ranked by their similarities (in descending order), since low-ranked books are not suitable for generating patterns. After testing three different proportions (10%, 33%, 50% which cover three major proportions within the 0% to 50% range), we decided to use 33% (or 280,066 pairs of most similar books) because it produced a recommendation list of manageable size (which avoids difficulties in analysis). For collaborative recommendation, we used all the transactions in the purchase history of each customer as input. For hybrid recommendation, we combined both content and collaborative association rules to form hybrid association rules.

Using this method with a minimum support value equal to 0.1%, we generated 476,990 valid content association rules that produced 6118 distinct rules heads, and 11,127 valid purchase association rules that produced 1450 distinct rule heads. The validity of an association rule was determined using the Chi-square test for statistical independence. We had much fewer purchase association rules than content association rules because customers usually purchased popular books (in terms of sales) while unpopular books seldom appeared in customers' purchase histories. As a result, most purchase association rules contained popular books in their rule heads while unpopular books seldom (or never) appeared in rule heads.

High-degree association retrieval. For high-degree association retrieval, we followed the method described herein previously and developed several stored procedures to perform spreading activation using a Hopfield net algorithm in the database.

In relatively large graph settings, the Hopfield net spreading activation is highly intensive computationally as it takes a long time to converge and therefore is not practical in real recommender systems. Another problem is that when a graph contains dense links, which means the nodes in the graph are reachable from almost any other node, some very popular books will overwhelm the recommendation list, reducing the high-degree association retrieval to a method based only on general popularity. We solved these problems by limiting the scope of spreading activation at each iteration in the Hopfield net algorithm. In each iteration, we computed the association weights of all the nodes pointed to by the current active nodes. Rather than activate all those nodes as described in the original Hopfield net algorithm, we limited the total number of active nodes by choosing the nodes having the 50 highest association weights. We also found that it was difficult to find an appropriate ε to perform sufficient but not overwhelming iterations. We altered the stopping criterion to generate acceptable results as follows.

Modified stopping criteria:

$$\sum \mu_j(t+1) - \sum \mu_j(t) \quad \varepsilon \times t$$

According to the modified stopping criteria, the convergence limit increased with the number of iterations. This way, with sufficiently small ε, the algorithm would not stop too early unless a reasonable size of recommendations had

TABLE 4. Average precision and recall for each configuration.

Measure	Precision			Recall		
Approach—method	Direct retrieval	Association mining	High-degree association retrieval	Direct retrieval	Association mining	High-degree association retrieval
Content-based	1.85%	1.14%	1.23%	6.51%	3.98%	3.46%
Collaborative	2.35%	**3.49%**	2.65%	13.20%	**12.80%**	15.00%
Hybrid	**2.98%**	2.29%	**3.39%**	**14.90%**	7.95%	**16.70%**

been obtained. When the program stopped at the *n*th iteration, we used the association weights of the nodes pointed to by active nodes in the (n − 1)th iteration, and chose the top 50 book nodes as recommendations.

Empirical Evaluation

We conducted an empirical evaluation to compare different recommendation approaches (content-based, collaborative, hybrid) and different proposed analysis methods (direct retrieval, association mining, high-degree association retrieval). With three approaches, each implemented by three analysis methods, we had nine configurations for comparison.

Procedure and parameters. To compare the effectiveness of the recommendations, we conducted a hold-out test on each configuration using information for 287 customers from our database. This sample represented the population of customers for whom the three recommendation methods (direct retrieval, association mining, high-degree association retrieval) could generate at least 50 recommendations. Then we employed the three methods to generate recommendations using three approaches (content, collaborative, hybrid). For each customer, we retrieved the list of books he/she previously had purchased. Each list was ordered by the purchase date. One half of the purchases (later purchases) were treated as "future" purchases and the other half (earlier purchases) were used to generate recommendations. We compared the recommended books with the "future" purchases to compute precision and recall measurements to evaluate different configurations. These metrics are defined as follows.

$$\text{Precision} = \frac{\text{Number of recommended books that match with the future purchases}}{\text{Total number of recommended books}}$$

$$\text{Recall} = \frac{\text{Number of recommended books that match with the future purchases}}{\text{Total number of books in the future purchases}}$$

Intuitively, precision captures the accuracy of our recommendations in predicting customers' purchase behavior while recall indicates the relevance of our recommendations to customers' personal interests. Table 4 shows the average precision and recall for each configuration. The highest precision (3.49%) was obtained in the collaborative–association mining configuration while the highest recall (16.7%) was obtained in the hybrid–high-degree association retrieval configuration. Not surprisingly, the precision and recall of all nine configurations in the hold-out test were very low, because there was often a gap between customers' interests and their actual purchase behaviors. The books purchased in the first half of a customer's purchase history might not have fully determined the books to be purchased in the second half of the purchase history. Many other factors that affect customer's purchase behavior cannot be captured by recommender systems.

Comparative study and results. We conducted 36 pairwise t tests to compare all pairs of the nine configurations. Table 5 shows the p-value of *t* tests on precision and recall. The upper portion of this table shows the results of testing the differences between different approaches. The lower portion shows the results of comparing the different methods. For example, when we compared precision measures between content-based recommendation and collaborative recommendation generated by the direct retrieval method, we found that the p-value was 0.036 and the sign was negative (−). This means that the precision obtained by the content-based approach was significantly smaller than that by the collaborative approach.

When comparing hybrid and content-based approaches, we found that a hybrid approach achieved significantly higher precision in all three methods. When comparing hybrid and collaborative approaches, we found that a hybrid approach achieved significantly higher precision than the direct retrieval method and the high-degree association retrieval method. Also, a hybrid approach achieved significantly higher recall than a content-based approach across all three methods and a significantly higher recall than a collaborative approach in direct retrieval and high-degree association retrieval methods. In summary, a hybrid approach performed better than either the content-based or the collaborative approach in most comparisons (five out of six comparisons in precision, five out of six comparisons in recall).

Regarding the differences among methods, we found that exploring high-degree associations did not consistently achieve higher precision. In only two comparisons (out of

TABLE 5. p-Values of pairwise t tests on precision and recall across different methods (alpha = 5%).

t-Test results by approaches (precision)[a]

Approaches	Direct retrieval	Association mining	High-degree association retrieval
Content-based vs. collaborative	0.036 (<)	0.000 (<)	0.000 (<)
Content-based vs. hybrid	0.000 (<)	0.000 (<)	0.000 (<)
Collaborative vs. hybrid	0.000 (<)	0.000 (>)	0.010 (<)

t-Test results by approaches (recall)[a]

Approaches	Direct retrieval	Association mining	High-degree association retrieval
Content-based vs. collaborative	0.000 (<)	0.000 (<)	0.000 (<)
Content-based vs. hybrid	0.000 (<)	0.000 (<)	0.000 (<)
Collaborative vs. hybrid	0.000 (<)	0.000 (>)	0.010 (<)

t-Test results by methods (precision)[b]

Methods	Content-based	Collaborative	Hybrid
Direct retrieval vs. association mining	0.009 (>)	0.001 (<)	
Direct retrieval vs. high-degree association	0.000 (>)	0.001 (<)	
Association mining vs. high-degree association	0.374 (<*)	0.016 (>)	

t-Test results by methods (recall)[b]

Methods	Content-based	Collaborative	Hybrid
Direct retrieval vs. association mining	0.002 (>)	0.387 (>*)	0.000 (>)
Direct retrieval vs. high-degree association	0.000 (>)	0.000 (<)	0.002 (<)
Association mining vs. high-degree association	0.249 (>*)	0.061 (<*)	0.000 (<)

[a] A less-than sign (<) means that the approach on the left yields a smaller precision or recall value than the approach on the right. A greater-than sign (>) means that the approach on the right yields a smaller precision or recall value than the approach on the left. An asterisk (*) means that the t-test result was not significant at a 95% confidence level.

[b] A less-than sign (<) means that the method on the left yields a smaller precision or recall value than the method on the right. A greater-than sign (>) means that the method on the right yields a smaller precision or recall value than the method on the left. An asterisk (*) means that the t-test result was not significant at a 95% confidence level.

six) did we find that high-degree association achieved significantly higher precision than other methods (direct retrieval vs. high-degree association retrieval—collaborative ($p = 0.001$); association mining vs. high-degree association retrieval—hybrid ($p = 0.009$)). Also, high-degree association did not consistently achieve higher recall. Only in three comparisons (out of six) did we find that high-degree association achieved significantly higher recall than other methods [direct retrieval vs. high-degree association retrieval, collaborative ($p = 0.000$) and hybrid ($p = 0.002$); association mining vs. high-degree association retrieval, hybrid ($p = 0.000$)].

Discussion

Our evaluation of the graph-based e-commerce recommendation system showed that a hybrid approach performed significantly better than a collaborative approach. This result confirmed our belief that combining content information and collaborative information would achieve better recommendations than using collaborative information alone and would provide an alternative recommendation approach to the many existing online businesses that primarily use a collaborative approach. Regarding the three recommendation methods (direct retrieval, association mining, .high-degree association), none significantly outperformed the others in terms of precision and recall. Contrary to our expectation, high-degree associations did not obtain a better recall than using direct retrieval. Direct retrieval achieved significantly better precision in three out of six comparisons. Although direct retrieval does not require advanced data processing such as association mining and Hopfield net spreading activation, it still achieved results comparable to those of other methods. From our nine configurations, we concluded that direct retrieval using a hybrid approach is the best system configuration because it achieved precision and recall comparable to the best values obtained, but did not involve an expensive computation such as association mining and high-degree association.

The direct retrieval method's high performance can be explained by the density of the data set we used. The data set contained 9695 books, 2000 customers, and 18,771

transactions, which means that on average, each customer purchased 9.4 books, and each book was purchased by 1.9 customers. In fact, the data set was formed by extracting the relevant data for the top 2000 active customers from the database of the online bookstore. We believe that when the average number of books purchased by each customer is small (as when the whole customer population is used as the data set to also include less active customers), the graph would become quite sparse and the high-degree association method might show its strength for solving the sparsity problem by exploring transitive neighborhoods. From our experiment, we can also conclude that association mining achieved the best precision when using the collaborative recommendation approach.

Conclusions

Recommender systems have been applied in different applications to deal with product information overload and to help locate useful information. Their data representations usually are tailored for their specific problems and hence are not flexible enough for use in other applications. In this article, we have described a generic graph model for product recommendation. Customers, products, and transactions were modeled in an extended two-layer graph. To demonstrate the comprehensiveness and flexibility of this model, we experimented with three methods (direct retrieval, association mining, high-degree association retrieval) of making recommendations using three approaches (content-based, collaborative, hybrid). Evaluation results showed that a hybrid recommendation approach achieved better performance than the collaborative approach or the content-based approach. However, the high-degree association method did not perform significantly better than the association mining method or the direct retrieval method because of the high density of the data set we used. We believe that our model is flexible because it can represent different combinations of product, customer, and transaction information and it has the potential to accommodate many graph search techniques for making recommendations. To further validate the current experimental results, we will extend the evaluation study to include frequently used recommendation data sets and directly compare our proposed graph-model-based approaches with approaches employed in previous studies. Other future work includes the use of other graph search algorithms as well as other data mining techniques, the use of a sparse data set to further test the strength of the high-degree association retrieval method and a complete subject evaluation on the usability of our system.

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 assistance during the project. We would also like to thank Thian-Huat Ong and Hui Liu for their involvement in system design and development.

References

Adomavicius, G., & Tuzhilin, A. (2001a). Extending recommender systems: A multidimensional approach. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-01), Workshop on Intelligent Techniques for Web Personalization (ITWP2001), Seattle, Washington, August 4–6.

Adomavicius, G., & Tuzhilin, A. (2001b). Using data mining methods to build customer profiles. IEEE Computer, 34(2), 74–82.

Aggarwal, C.C., Wolf, J.L., Wu, K.-L., & Yu, P.S. (1999). Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In Proceedings of the Fifth ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'99) (pp. 201–212), San Diego, CA, August 15–18.

Aggarwal, C.C., & Yu, P.S. (2001). Mining associations with the collective strength approach. IEEE Transactions on Knowledge and Data Engineering, 13, 863–873.

Ansari, A., Essegaier, S., & Kohli, R. (2000). Intemet recommendations systems. Joumal of Marketing Research, 37(3), 363–375.

Baeza-Yates, R., & Gonnet, G. (1996). Fast text searching for regular expressions or automaton searching on tries. Joumal of the ACM, 43(6), 915–936.

Balabanovic, M., & Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. Communications of the ACM, 40(3), 66–72.

Basu, C., Hirsh, H., & Cohen, W. (1998). Recommendation as classification: Using social and content-based information in recommendation. In Proceedings of the Fifteenth National Conference on Artificial Intelligence (pp. 714–720), Madison, Wisconsin, July 26–30.

Breese, J.S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (pp. 43–52), Madison, WI: Morgan Kaufmann.

Burke, R. (2000). Semantic ratings and heuristic similarity for collaborative filtering. In Proceedings of the Seventeenth National Conference on Artificial Intelligence, Austin, Texas, July 30–August 3.

Chen, H., & Lynch, K.J. (1992). Automatic construction of networks of concepts characterizing document databases. IEEE Transactions on Systems, Man, and Cybernetics, 22, 885–902.

Chen, H., & Ng, D.T. (1995). An algorithmic approach to concept exploration in a large knowledge network (automatic thesaurus consultation): Symbolic branch-and-bound search vs. Connectionist Hopfield net activation. Joumal of the American Society for Information Science, 46, 348–369.

Chau, M., Zeng, D., Chen, H., Huang, M., & Hendriawan, D. (2003). Design and evaluation of a multi-agent collaborative Web mining system. Decision Support Systems (DSS), Special Issue on Web Retrieval and Mining, 35(1), 167–183.

Church, K. (1988). A stochastic parts program and noun phrase parser for unrestricted text. In Proceedings of the Second Annual Conference on Applied Natural Language Parsing ACL (pp. 136–143), Austin, TX.

Claypool, M., Gokhale, A., Miranda, T., Mumikov, P., Netes, D., & Sartin, M. (1999). Combining content-based and collaborative filters in an online newspaper. In Proceedings of the ACM SIGIR Workshop on Recommender Systems, Berkeley, California, August 19.

Cohen, E., Datar, M., Fujiwara, S., Gionis, A., Indyk, P., Motwani, R., Ullman, J.D., & Yang, C. (2001). Finding interesting associations without support pruning. IEEE Transactions on Knowledge and Data Engineering, 13(1), 64–78.

Condliff, M.K., Lewis, D.D., Madigan, D., & Posse, C. (1999). Bayesian mixed-effects models for recommender systems. In Proceedings of the ACM SIGIR Workshop on Recommender Systems, Berkeley, California, August 19.

Fu, X., Budzik, J., & Hammond, K. (2000). Mining navigation history for recommendation. In Proceedings of the 2000 International Conference on Intelligent User Interfaces (pp. 106–112), New Orleans, Louisiana, January.

Fumer, J. (2002). On recommending. Journal of the American Society for Information Science and Technology, 53(9), 747–763.

Goldberg, D., Nichols, D., Oki, B., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12), 61–70.

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval, 4(2), 133–151.

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

Huang, Z., Chung, W., Ong, T.-H., & Chen, H. (2002). A graph-based recommender system for digital library. In Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 65–73), Portland, Oregon, July 14–18.

Lieberman, H. (1995). Letizia: An agent that assists web browsing. In Proceedings of the ntemational Joint Conference on Artificial Intelligence (pp. 924–929), Montreal, Canada.

Lin, W., Alvarez, S.A., & Ruiz, C. (2002). Eficient adaptive-support association rule mining for recommender systems. Data Mining and Knowledge Discovery, 6, 83–105.

Manber, U., & Myers, G. (1993). Suffix arrays: A new method for on-line string searches. SIAM Journal on Computing, 22(5), 935–948.

Mcdonald, D., & Ackerman, M. (2000). Expertise recommender: A flexible recommendation system and architecture. In Proceedings of the ACM Conference on Computer Supported Cooperative Work (pp. 231–240), Philadelphia, PA, December 2–6.

Mirza, B.J. (2001). Jumping connections: A graph-theoretic model for recommender systems. Retrieved February 1,2003, from http://scholar-.lib.vt.edu/theses/available/etd-02282001-175040/unrestricted/etd.pdf

Mladenic, D. (1996). Personal Webwatcher: Implementation and design. Technical Report, US-DP-7472, Department of Intelligent Systems, J. Stefan Institute, Slovenia.

Mobasher, B., Dai, H., Luo, T., Nakagawa, M., Sun, Y., & Wiltshire, J. (2000). Discovery of aggregate usage profiles for Web personalization. In Proceedings of the Workshop on Web Mining for E-Commerce—Challenges and Opportunities, Boston, Massachusetts, August 20.

Mooney, R., & Roy, L. (2000). Content-based book recommending using leaming for text categorization. In Proceedings of the Fifth ACM Conference on Digital Libraries (pp. 195–204), San Antonio, TX, June 2–7.

Mostafa, J., Mukhopadhyay, S., Lan, W., & Palakal, M. (1997). A multilevel approach to intelligent information filtering: Model, system, and evaluation. ACM Transactions on Information Systems, 15(4), 368–399.

Nasraoui, O., Frigui, H., Joshi, A., & Krishnapuram, R. (1999). Mining Web access logs using relational competitive fuzzy clustering. In Proceedings of the Eighth Intemational Fuzzy Systems Association World Congress—IFSA 99, Hsinchu, Taiwan, August.

Ong, T., & Chen, H. (1999). Updateable pat-tree approach to Chinese key phrase extraction using mutual information: A linguistic foundation for knowledge management. In Proceedings of the Second International Conference of Asian Digital Libraries (pp. 63–84), Taipei, Taiwan, November 8–9.

Pazzani, M. (1999). A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review, 13(5), 393–408.

Pazzani, M., & Billsus, D. (1997). Leaming and revising user profiles: The identification of interesting Web sites. Machine Leaming, 27, 313–331.

Polloc, S. (1988). A rule-based message filtering system. ACM Transactions on Information Systems, 6(3), 232–254.

Resnick, P., & Varian, H. (1997). Recommender systems. Communications of the ACM, 40(3), 56–58.

Salton, G. (1989). Automatic text processing: The transformation, analysis and retrieval of information by computer. Reading, MA: Addison Wesley.

Sarwar, B., Konstan, J., Borchers, A., Herlocker, J., Miller, B., & Riedl, J. (1998). Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system. In Proceedings of the ACM Conference on computer Supported Cooperative Work (CSCW) (pp. 345–354), Seattle, Washington, November 14–18.

Sarwar, B.M., Karypis, G., Konstan, J.A., & Riedl, J.T. (2001). Item-based collaborative filtering recommendation algorithms. In Proceedings of the Tenth International World Wide Web Conference (pp. 285–295), Hong Kong, China, May 1–5.

Schafer, J., Konstan, J., & Riedl, J. (2001). E-commerce recommendation applications. Data Mining and Knowledge Discovery, 5(1–2), 115–153.

Schein, A.I., Popescul, A., Unger, L.H., & Pennock, D.M. (2002). Methods and metrics for cold-start recommendations. In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR2002) (pp. 253–260), Tampere, Finland.

Schwab, I., Phol, W., & Koychev, I. (2000). Leaming to recommend from positive evidence. In Proceedings of the Intemational Conference on Intelligent User Interfaces (pp. 241–247). New Orleans, LA: ACM Press, January 9–12.

Singh, M.P., Yu, B., & Venkatraman, M. (2001). Community-based service location. Communications of the ACM, 44(4), 49–54.

Terveen, L., Hill, W., Amento, B., Mcdonald, D., & Creter, J. (1997). Phoaks: A system for sharing recommendations. Communications of the ACM, 40(3), 59–62.

Watters, C., & Wang, H. (2000). Rating news documents for similarity. Joumal of the American Society for Information Science and Technology, 51(9), 793–804.