A Unified Recommendation Framework Based on Probabilistic Relational Models*

Zan Huang, Daniel D. Zeng, Hsinchun Chen
Department of Management Information Systems
The University of Arizona
{zhuang, zeng, hchen}@eller.arizona.edu

Abstract

Recommender systems are being increasingly adopted in various e-commerce applications. A wide range of recommendation approaches have been developed to analyze past consumer-product interactions, consumer attributes, and product attributes to predict future sales. In this paper we propose a unified recommendation framework based on *probabilistic relational models* (PRMs). This framework includes most of existing recommendation approaches, such as collaborative filtering, content-based, demographic filtering, and hybrid approaches, as special cases. Recently developed in the machine learning community, PRMs aim to study the relational patterns within a database containing multiple interlinked data tables using a statistical model that describes probabilistic dependencies between attributes in the domain. We extended the original PRMs in order to capture relational data patterns that are important for recommendation. We also specialized the algorithm for learning PRMs in dependency model construction and parameter estimation to exploit the special characteristics of the recommendation problem. Through an experimental study, we demonstrate that the proposed framework not only conceptually unifies existing recommendation approaches but also allows the exploitation of a wider range of relational data patterns in an integrated manner, leading to improved recommendation performance.

1. Introduction

Recommender systems are being widely used in many application settings to suggest products, services, and information items to potential consumers. A wide range of companies such as *Amazon.com*, *Half.com*, *CDNOW*, *J.C. Penney*, and *Procter & Gamble* have successfully deployed recommendation technologies to increase Web and catalog sales and improve customer loyalty [25]. A variety of recommendation approaches have been developed in the Artificial Intelligence and Information Retrieval communities [1, 19, 22]. Most of theses approaches take as input three types of data: product attributes, consumer attributes, and interactions between consumers and products (such as purchases and ratings). As output, they predict future or unobserved interactions as recommendations. Depending on the input data these approaches can be roughly categorized into content-based (using product attributes and the interaction data), collaborative filtering (using the interaction data only), demographic filtering (using consumer attributes and the interaction data), and hybrid approaches (using multiple types of input data) [12, 19, 22].

Conceptually, the recommendation problem is concerned with the relationships between consumers and products. As such, it can be viewed as a special case of the *relational learning* problem [4]. Recent years have seen significant interest and development in the area of relational learning, which focuses on identifying relational patterns within a database containing multiple interlinked data tables. Applying the relational learning framework, one can argue that a recommendation model takes a (portion of the) database containing multiple related tables regarding consumers, products, and their interactions as input to predict unobserved entries in the consumer-product interaction table.

The main objective of this paper is to establish the connection between the recommendation problem and the relational learning framework through the application of a recently developed statistical relational learning method called *probabilistic relational models* (PRMs) in the recommendation context. We

* An earlier version of this paper was presented in the Fourteenth Annual Workshop on Information Technologies and Systems (WITS 2004) and received the Best Paper Award.

extend the original PRMs to meet the unique computational challenges of the recommendation task. We show that existing recommendation approaches can be conceptualized as special cases under this framework. We also demonstrate the improved recommendation performance achieved by the unified framework using real-world e-commerce datasets.

2. Relational Learning and Probabilistic Relational Models

Relational learning or *multirelational learning* [4] extends standard data mining that learns from attributes of independent entities stored in a single database table to extract patterns from multiple related tables. The assumption that the data objects are independent from each other is dropped in relational learning. In fact, linkages between data objects are of central interest in relational learning. Examples of relational learning applications include link prediction, link-based clustering, social network modeling, and object identification.

Probabilistic relational models (PRMs) are the main formal approach that has been developed for relational learning [15, 21]. A PRM is defined for a particular database, or formally a *relational schema*. A relational schema R describes a set of classes (tables in the database) $\boldsymbol{X}$. Each $X \in \boldsymbol{X}$ is associated with a set of *descriptive attributes* (standard table attributes) $\boldsymbol{A}(X)$ and a set of *reference slots* (foreign keys) $\boldsymbol{R}(X)$. We denote the attribute A of class X as $X.A$ and the reference slot ρ of X as $X.\rho$, where ρ denotes a function from Domain[ρ] = X to Range[ρ] = Y. PRMs with *existence uncertainty* [6] are able to model the existence of certain records in the data tables. Under this extension, a class X of interest can be modeled as an *undetermined* class by introducing a special *existence* attribute $X.E$ whose values are from $V(E)$ = {*true*, *false*}, with *true* indicating the particular object of class X exists and *false* indicating nonexistence. For each reference slot ρ we define an inverse reference slot ρ^{-1}, mapping from Range[ρ] = Y to Domain[ρ] = X. A slot chain τ is defined as $\tau = \rho_1,\ldots,\rho_k$, for all i, Range[ρ_i] = Domain[ρ_{i+1}]. Through slot chains dependencies between the attributes of related data objects can be explored.

A PRM is an extension of *Bayesian networks* for describing probability distributions over a database. A PRM Π contains a qualitative component S, an acyclic graph that describes the statistical dependency structure of descriptive attributes linked through slot chains, and a quantitative component Θ_S that represents the set of parameters characterizing the conditional probability distributions. Formally, a PRM $\Pi = \langle S, \Theta_S \rangle$ for a *relational schema* $R = \langle \boldsymbol{X}, \boldsymbol{A} \rangle$ defines for each class $X \in \boldsymbol{X}$ and each descriptive attribute $A \in \boldsymbol{A}(X)$, a set of parents $Pa(X.A)$, and a conditional probability distribution that represents $P(X.A \mid Pa(X.A))$ [6]. A complete *instantiation I* for a PRM is defined as the set of objects in each class X and the values for each attribute and each reference slot of each object. With a complete instantiation a PRM can be learned by finding a PRM Π that best matches I. Similar to Bayesian network learning, a statistically motivated scoring function is used to evaluate each model with respect to the training data. A commonly used Bayesian scoring metric is given by $\log P(S \mid I) = \log P(I \mid S) + \log P(S) + C$, where $P(I \mid S)$ is the marginal likelihood $P(I \mid S) = \int P(I \mid S, \Theta_S) P(\Theta_S \mid S) d\Theta_S$. Standard hill-climbing greedy search algorithms can be employed to search for the optimal structural model S. With the optimal dependency structure, standard *maximum likelihood* parameter estimation can be performed to complete the model specification. Details on PRM learning can be found in [5, 6].

3. PRM-based Recommendation

The recommendation problem is an ideal application for relational learning as the linkages between consumers and products are the modeling focus. In fact one most successful recommendation approach, collaborative filtering, makes recommendation predictions only based on these linkages. Several papers have discussed how to apply PRMs to recommendation problems conceptually [6, 7]. A recent study evaluated an extended version of PRMs using movie rating data but did not show significant performance improvement of the PRM-based approach over existing recommendation algorithms [18]. The PRM framework can be relatively easily adapted to make rating-based recommendations, since the output field (ratings) of learning is a typical descriptive attribute value. However, applying the PRM framework to predict whether or not a transaction will take place (a binary prediction) is significantly harder. Part of the reason is associated with a fundamental difference between recommendation problems based on ratings and transactions. Consider a simple rating recommendation of either like (+1) or dislike (−1). There are three possible rating values for each consumer-product pair, +1, −1, and 0 (representing no observation of rating). Typical rating-based recommendation takes the observations with +1 and −1 ratings as training data to derive models to predict potential rating values for unobserved ratings. For recommendation with binary transaction data, however, the only information regarding a consumer-product pair is whether a transaction has taken place (1) or not (0). Within the relational learning context, this corresponds to a link (or relationship instance) prediction problem, as opposed to the attribute value prediction. In our study, we extend the original PRMs to deal with the link prediction problem for transaction-based recommendation and in turn provide a framework to unify existing recommendation algorithms for both rating-based and transaction-based recommendations.

In this section, we establish the connection between recommendation problem and relational learning. Using a book sales database as an example, Section 3.1 describes recommendation task from a relational learning perspective. Section 3.2 introduces an extension to the original PRM, motivated to evaluate similarities between sets required by the recommendation model. Section 3.3 presents a simplified parameter estimation procedure exploiting the characteristics of recommendation tasks. We demonstrate in Section 3.4 that existing recommendation approaches can be viewed as special cases of our unified framework.

3.1 Model Description

Figure 1 illustrates an example book sales database. Customer, Book, and Word are entity classes while Order and Occurrence are relationship classes. Customer and Book have 7 and 3 descriptive attributes, respectively, excluding their identifiers. Order contains two reference slots linking to Customer and Book while Occurrence contains book and word reference slots describing the occurrence of keywords in book content descriptions such as title and introduction. For recommendations, we model Order as an undetermined class by introducing a special descriptive attribute exist to indicate the existence of a sales transaction. In Figure 1, we only present existing transaction instances (with 1 assigned to the exist attribute) while for all other customer-book pairs the value 0 is implicitly assigned indicating the absence of the sale transaction.

Customer

customer	city	birthYear	education	vocation	sex	married	child
c1	taipei	1977	college	financial	f	yes	1
c2	kaohsuing	1968	high school	construction	m	no	0
c3	taipei	1982	college	student	m	no	0

Order

customer	book	exist
c1	b1	1
c1	b2	1
c2	b3	1
c3	b4	1

Book

book	publisher	translated	price
b1	p1	yes	130
b2	p1	yes	230
b3	p2	no	100
b4	p3	no	500

Occurrence

book	word
b1	w2
b1	w3
b2	w1
b3	w4
b3	w5
b4	w4
b4	w6

Word

word
w1
w2
w3
w4
w5
w6

Figure 1. An example book sales database

Using the PRM notation introduced in Section 2, the dependency structure S for a PRM defines the parents $Pa(X.A)$ for each attribute $X.A$. In our context, since we are only concerned with Order.exist, we only need to derive a partial dependency structure for Order.exist. Instead of searching for the complete model describing all probabilistic dependencies we focus on identifying attributes within the *Markov blanket* of Order.exist and search for an optimal model describing these variables and Order.exist. A Markov blanket refers to the parents, children, and other parents of the children of a node V in a Bayesian network model; it shields V from being affected by any node outside the blanket [2]. Potential attributes to be included into the Markov blanket of Order.exist can be derived from reference slots or slot chains. For example, [Order.customer].education could be a potential parent attribute representing the education level of the target customer. Long slot chains with inverse reference slots can bring in more complex attributes. For example, [Order.customer].[Order.customer]$^{-1}$.[Order.book].price represents the prices of the set of books bought by the target customer. If any of the reference slots in the chain involves a one-to-many mapping, such as [Order.customer]$^{-1}$ (indicating the function from a customer to his/her involved orders) in this example, the derived attribute will be a multi-valued attribute. For these attributes, aggregation operators such as *maximum, minimum, mode, average,* and *carnality* can be applied and aggregated single-valued attributes are then included into the dependency structure *S*.

3.2 Multi-set Operations

A PRM allows attributes to be derived separately from individual slot chains. For example, [Order.customer].[Order.customer]$^{-1}$.[Order.book].[Order.book]$^{-1}$.[Order.customer] represents the set of customers who bought at least one common book bought by the target customer (the customer neighbors) while [Order.book].[Order.book]$^{-1}$.[Order.customer] represents the set of customers who bought the target book. These two multi-valued attributes, with aggregation operations, could provide certain information regarding the likelihood for a transaction involving the target consumer and target book to occur. However, the original PRMs do not model information that can only be derived *jointly* from multiple attributes, which can play a critical role in recommendation. For example, the set similarity of the above two attributes (may be derived through the cardinalities of the intersection and union of the two attributes) describes the overlap of the target consumer's neighbors and the customers who bought the target book. Such information is essential for making user-based collaborative filtering recommendations.

To derive information jointly from multiple attributes, we propose to extend PRMs by introducing *multi-set operators*. A multi-set operator ϕ_k on k multi-valued attributes A_1, …, A_k that share the same domain $V(A_1)$ denotes a function from $V(A_1)^k$ to $V(A_1)$. Such multi-set operators include simple set operators such as *intersection* and *union* and more complex aggregation operators modeling value distributions [20]. By applying an aggregation operator after a multi-set operator, we can derive attributes from multiple multi-valued attributes and include them into the probabilistic dependency specification of a PRM. Our current study focuses on using binary multi-set operators that involve two attributes.

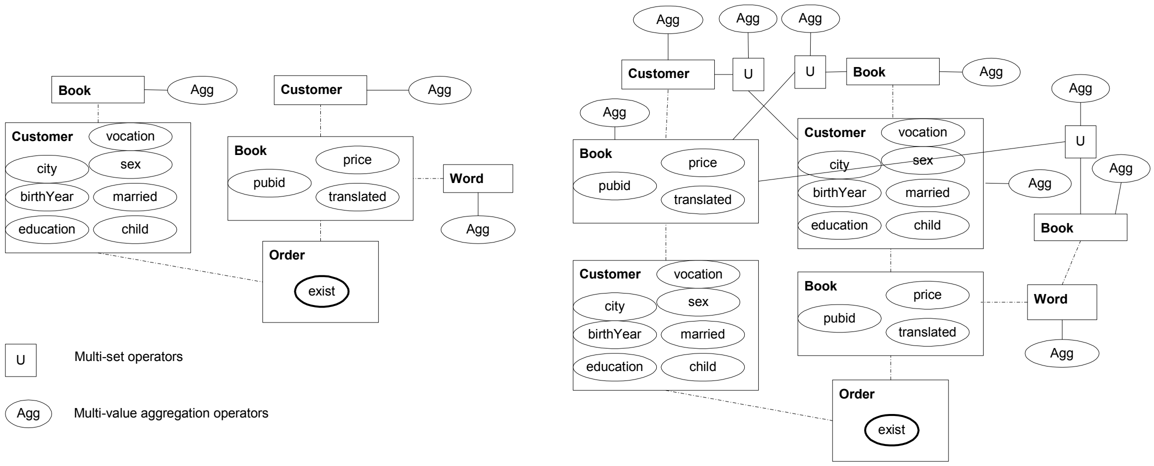
3.3 Learning Process

An important challenge for PRM learning is that there are an infinite number of potential attributes that could be derived through slot chains and the newly introduced multi-set operators. The standard approach to address this issue is an iterative expanding heuristic structure search algorithm [5]. Under this approach, the length of the slot chains is constrained while searching for the optimal PRM. The allowable slot chain length controls the complexity of the model.

We applied standard search-and-scoring procedure for Bayesian network learning to search the optimal partial dependency structure involving Order.exist as discussed in Section 3.1. Once the optimal local PRM dependency structure for a particular slot chain length is determined, standard parameter estimation procedures can be applied to derive predictive models of the value of Order.exist. In our current study, a Naïve Bayesian algorithm for binary prediction [2] was applied to estimate the purchase probability P(Order.exist = 1 | relevant attributes of Order.exist) for unobserved customer-book pairs. Various types of recommendations can then be generated based on such purchase probability estimates.

3.4 Recommendation Models under PRM

We now examine existing recommendation approaches in light of our unified PRM framework. For illustration purposes, we use the same book sales database example. In Figure 2, attributes in circles, including single-valued and aggregated multi-valued attributes, are potential relevant attributes of Order.exist in the dependency model. Model (a) includes attributes derived through slot chains with the maximum length of 3 (e.g., city of the target customer [Order.customer].city and number of customers who bought the target book *cardinality*{[Order.book].[Order.book]$^{-1}$.[Order.customer]}). Such a model corresponds to typical purchase prediction models that involve customer demographic attributes, number of observed customer purchases, book attributes, and past book sales volumes.



(a) Maximum length of slot chains: 3 (b) Maximum length of slot chains: 5

Figure 2. PRM recommendation model with maximum slot chain lengths of 3 and 5

Model (b) allows for slot chains of the maximum length of 5. As the model gets more complicated, many interesting attributes that correspond to existing recommendation approaches appear. As explained in Section 3.2, *cardinality*{*intersection*{[Order.customer].[Order.customer]$^{-1}$.[Order.book].[Order.book]$^{-1}$.[Order.customer], [Order.book].[Order.book]$^{-1}$.[Order.customer]}} represents the number of the target customer's neighbor customers who have bought the target book, which provides essential information for the standard user-based collaborative filtering algorithm. Similarly, *cardinality*{*intersection* {[Order.customer].[Order.customer]$^{-1}$.[Order.book], [Order.book].[Occurrence.book]$^{-1}$.[Occurrence.word].[Occurrence.word]$^{-1}$.[Occurrence.book]}} represents the number of books bought by the target customer that contain words appearing in the target book, which provides essential information for content-based recommendation approaches. The PRM estimated based on model (b) could potentially be a "hybrid" recommendation based on multiple algorithmic implementations of several recommendation approaches. As the maximum slot chain length increases, the model becomes more complicated by introducing indirect customer/book neighbors and their associated attributes, which in principle correspond to graph-based recommendation algorithms that account for transitive associations among customers and books leading to better recommendation performances (e.g., [11]). However, a larger maximum slot chain length also leads to a dramatically larger search space, making the model estimation and prediction process much more computationally intensive.

4. An Experimental Study

In this section we report an experimental study evaluating the effectiveness of the proposed recommendation framework. In the first part of the study, we explore how well the PRM-based framework can automatically synthesize data patterns traditionally exploited by different recommendation approaches separately. To further demonstrate the advantage of our proposed framework, in the second part of the study we compare the performance of the proposed PRM-based framework directly with existing collaborative filtering algorithms by limiting its input to sales transaction table only, as used by the collative filtering algorithms.

4.1 Data

We used two e-commerce datasets in our experimental study: a retail dataset provided by a leading U.S. online clothing merchant and a book dataset provided by a Taiwan online bookstore. The retail dataset contained 3 months of transaction data that contain about 16 million transactions (household-product pairs) involving about 4 million households and 128,000 products. The book dataset contained 3 years of transactions of a sample of 2,000 customers. There were about 18,000 transactions and 9,700 books involved in this dataset.

For our experiments, we used samples from these two datasets. We included consumers who had purchased 5 to 100 products for meaningful testing of the recommendation algorithms. This range constraint resulted in 851 consumers for the book dataset and about 1.2 million households for the retail dataset. To keep the sample size close to each other, for the retail dataset we randomly selected 1,000 from the 1.2 million households. The details about the final samples are shown in Table 1. The statistics of the complete datasets are also reported within the parentheses.

Dataset	# of Consumers	# of Products	# of Transactions	Density Level*	Avg. # of purchases per consumer	Avg. sales per product
Retail	1,000 (~4 million)	7,328 (~128,000)	9,332 (~16 million)	0.13% (~0.0031%)	9.33 (~4)	1.27 (~125)
Book	851 (~2,000)	8,566 (~9,700)	13,902 (~18,000)	0.19% -0.09%	16.34 (~9)	1.62 (~1.86)

Table 1. Characteristics of the datasets (* the *density level* of a dataset is defined as the percentage of the elements valued as 1 in the interaction matrix).

The book dataset contained detailed information describing the consumers (e.g., city, year of birth, education) and books (e.g., price, publisher, authors, title, keyword, introduction). We preprocessed the data and transformed it into a normalized database. The schema of the database is shown in Figure 1 of Section 3.1. For the word occurrence table we included indexed phrases in book titles and keywords. We used this dataset in the first part of the study to evaluate the capability of the proposed framework to generate high-quality hybrid recommendations based on collaborative, content-based, and demographic-based data patterns. The retail dataset was quite limited, only containing the sales transaction table. This dataset was used together with the sales transaction table of book dataset for the second part of the study to compare the performances of the PRM-based framework with those of existing collaborative filtering algorithms.

4.2 Evaluation Procedure and Performance Measures

While an ideal evaluation of a recommendation will involve assessment given by customers who actually experienced the recommended products or services, this type of evaluation is typically difficult and prohibitively expensive. In the recommendation algorithm research literature, a hold-out test procedure is commonly adopted to evaluate algorithm performances. The idea is to withhold a portion of the transaction data as the testing set and use the remaining transactions (referred to as the training set) to generate recommendations. The recommendations are then compared with the actual transactions in the testing to derive performance measures.

In our study, we selected 20% of each consumer's interactions (the latest) to form the testing set and designated the remaining 80% (earlier interactions) to be the training set. We focused on the well-studied Top-N recommendation task, in which a ranked list of N products is recommended to each consumer. For each consumer, the recommendation quality was measured based on the number of *hits* (recommendations that matched the products in the consumer's testing set) and positions of the hits in the ranked list. We adopted the following recommendation quality metrics from the literature regarding the relevance, coverage, and ranking quality of the ranked list recommendation (e.g., [1]): Precision: $P_c = \frac{\text{Number of } hits}{N}$, Recall: $R_c = \frac{\text{Number of } hits}{\text{Size of } c's \text{ testing set}}$, F Measure: $F_c = \frac{2 \times P_c \times R_c}{P_c + R_c}$, and Rank Score: $RS_c = \sum_j \frac{q_{cj}}{2^{(j-1)/(h-1)}}$, where j is the index for the ranked list; h is the viewing *halflife* (the rank of the product on the list such that there is a 50% chance the user will purchase that product); $q_{cj} = \begin{cases} 1, \text{ if } j \text{ is in } c\text{'s testing set,} \\ 0, \text{ otherwise} \end{cases}$. For precision, recall, and F measure, an average value over all consumers tested was adopted as the overall metric for the algorithm. For the rank score, an aggregated rank score RS for all consumers tested was derived as $RS = 100 \frac{\sum_c RS_c}{\sum_c RS_c^{\max}}$, where $RS_c^{\max}$ was the maximum achievable rank score for consumer *c* if all future purchases had been at the top of a ranked list. The precision, recall, and F measure are standard performance measures to estimate the relevance and coverage of the recommended items compared with the consumers' potential interests. The rank score measure was proposed in [1] and adopted in many follow-up studies (e.g., [3, 9, 11]) to evaluate the ranking quality of the recommendation list. In our experiments, we set the number of recommendations to be 10 (N = 10) and the halflife for the rank score to be 2 (h = 2).

4.3 PRM-based Recommendations

In this section we report the experimental results with the complete book sales dataset under the proposed PRM-based recommendation framework. By setting the allowable slot chain length to be 5, we constructed PRM models over the attribute space as shown in Figure 2(b). As discussed in Section 3.4, subsets of attributes in the attribute space can be included to form recommendation models that correspond to different recommendation approaches including collaborative filtering, content-based, demographic filtering, and hybrid approaches. In our experiment, we categorized the attributes into content-based, demographic-based, and collaborative attributes. Content-based attributes are the ones involved with the Book and Occurrence tables. Demographic-based attributes are the ones involved with the Customer table. Collaborative attributes are the ones only involved with the Order table. Restricting on attributes of individual categories, we followed the learning procedure described in Section 3 to select the sets of relevant attributes and derived PRM models for different recommendation approaches, referred to as *PRM-Content*, *PRM-Demographic*, and *PRM-Collaborative*, respectively. We also derived an unrestricted model using the entire attribute space, referred to as *PRM-Complete*.

Model	ID	Attribute	Interpretation*
PRM-Content	A1	cardinality{intersection{[Order.customer].[Order.customer]$^{-1}$.[Order.book], [Order.book].[Occurrence.book]$^{-1}$.[Occurrence.word].[Occurrence.word]$^{-1}$.[Occurrene.book]}}	Number of books purchased by *c* that contain the words appearing in *b*
	A2	cardinality{intersection{[Order.book].[Occurrence.book]$^{-1}$.[Occurrence. word], [Order.customer].[Order.customer]$^{-1}$.[Order.book]. [Occurrence. book]$^{-1}$.[Occurrene.word]}}	Number of words in *b* that also appear in books purchased by *c*
PRM-Demographic	A3	cardinality{intersection{[Order.customer].[Customer.birthYear], [Order. book].[Order.book]$^{-1}$.[Order.customer].[Customer.birthYear]}}	Number of customers who purchased *b* and have the same range of year of birth as *c*
	A4	cardinality{intersection{ [Order.customer].[Customer.education], [Order. book].[Order.book]$^{-1}$.[Order.customer].[Customer.education]}}	Number of customers who purchased *b* and have the same level of education as *c*
PRM-Collaborative	A5	cardinality{intersection{ [Order.customer].[Order.customer]$^{-1}$.[Order. book], [Order.book].[Order.book]$^{-1}$.[Order.customer].[Order.customer]$^{-1}$. [Order.book]}}	Number of books purchased by *c* that have been bought by common customers with *b*
	A6	card{[Order.book].[Order.book]$^{-1}$.[Order.customer]}	Number of customers who bought *b* (sales volume of *b*)
PRM-Complete		A1, A4, A5, A6	

Table 2. Relevant attributes in PRM models (* the target customer is denoted by *c* and the target book is denoted by *b*).

Model	Precision	Recall	F Measure	Rank Score
PRM-Content	0.0142	0.0767	0.0227	5.4225
PRM-Demographic	0.0145	0.0778	0.0229	7.4946
PRM-Collaborative	0.0267	0.1354	0.0417	**11.1411**
PRM-Complete	**0.0313**	**0.1636**	**0.0493**	**12.0511**

Table 3. Recommendation performance measures of PRM models (boldfaced measures were not significantly different from the largest measure at the 5% significance level).

Table 2 presents the relevant attributes selected within each PRM model. The recommendation performance measures of the individual PRM models are presented in Table 3. The results show that the content-based and demographic filtering recommendation achieved similar precision, recall, and F measure. The demographic filtering recommendation performed better than the content-based recommendation in placing the correct recommendations to higher ranks in the ranked list, as indicated by its higher rank score. The collaborative filtering model achieved much better recommendation performances than the content-based and demographic filtering models, confirming the observations in the literature. The results show a significant improvement of the *PRM-Complete* model over the *PRM-Collaborative* model. Our experiments also show that removing any single attribute (except the demographic attribute A4) in the *PRM-Complete* model would lead to degraded recommendation performances. Overall, these results indicate that our proposed PRM-based recommendation framework successfully identified the unique set of attributes across content-based, demographic-based, and collaborative data patterns, which lead to better recommendation performances.

4.4. Collaborative Filtering Performance Comparison

To further demonstrate the potential advantage of the proposed recommendation framework in the second part of our study, we compared our proposed framework with a wide range of existing collaborative filtering algorithms using the retail and book sales transaction datasets.

ID	Attribute	Interpretation*
X1	cardinality{[Order.book].[Order.book]$^{-1}$.[Order.customer]}	Sales volume of *b*
X2	cardinality{intersection{ [Order.customer].[Order.customer]$^{-1}$.[Order.book], [Order.book].[Order.book]$^{-1}$.[Order.customer].[Order.customer]$^{-1}$.[Order.book]}}	Number of books purchased by *c* that have been bought by common customers with *b*
X3	cardinality{intersection{ [Order.customer].[Order.customer]$^{-1}$.[Order.book], [Order.book]. [Order.book]$^{-1}$. [Order.customer].[Order.customer]$^{-1}$.[Order.book]. [Order.book]$^{-1}$.[Order.customer] .[Order.customer]$^{-1}$.[Order.book]}	Number of transitive neighbors of *b* (the neighbors of *b*'s neighbors) that have been purchased by *c*

Table 4. Attribute in the attribute space for collaborative filtering recommendation (* the target customer is denoted by *c* and the target book is denoted by *b*).

For this experiment we increased the allowable slot chain length to 7 because with only the Order class the attribute space is greatly reduced. Three attributes were included in the models for the two datasets, which are presented in Table 4. X3 is a newly introduced attribute constructed through a slot chain of length 7. It involves the concept of transitive neighbors of the target book, which refers to the books purchased by the customers who purchased books that were purchased together with the target book previously. Intuitively, it represents a relatively long path in a consumer-product graph induced by the sales transactions: *b*—*c'*—*b'*—*c''*—*b''*. The data pattern represented by X3 corresponds to transitive associations studied in several graph-based collaborative filtering algorithms [11, 13, 16].

For the book dataset, the PRM model did not change from the one reported previously in Table 2, which consists of X1 and X2 in Table 4. A different model was derived for the retail dataset, which consists of X1 and X3. The model difference might be related to the different data patterns within the two datasets: popularity-based transactions may be more frequent in the book dataset, while transitive associations could potentially lead to possible transactions in the retail dataset.

For comparison purposes, we reported in Table 5 the performance measures for the PRM models for the book and retail datasets as well as the performances of 7 collaborative filtering algorithms on the same datasets obtained from our previous research [14]. These collaborative filtering algorithms include: user-based neighborhood algorithm [1], item-based neighborhood algorithm [3, 24], dimensionality reduction algorithm [8, 23], generative model algorithm [10, 26], two graph-based algorithms including the spreading activation algorithm [11] and link analysis algorithm [13], and a naïve recommendation based entirely on observed sales volume of the products.

Based on the results shown in Table 5, the PRM clearly dominated all other collaborative filtering algorithms in all four performance measures for the retail dataset. The fact that the attribute X3 was included in the PRM model for the retail datasets was consistent with the relatively good performance of the graph-based algorithms (spreading activation and link analysis algorithms), which also exploits transitive associations among consumers and products. The PRM's superior performance over the graph-based algorithms may be attributed to its nature of supervised learning and the capability to combine transitive association (X3) with the book sales information (X1). For the book dataset, PRM also achieved the best performances, although not dominating all other algorithms. The naïve recommendation based on sales volume had surprisingly good performance for the book dataset while the graph-based algorithms did not show significant improvement over other algorithms. These results may provide explanation for the PRM to be able to deliver the best performance with relatively simple data patterns (X1 and X2).

Algorithm	Retail				Book			
	Precision	Recall	F Measure	Rank Score	Precision	Recall	F Measure	Rank Score
Naïve	0.0062	0.0326	0.0100	1.3889	**0.0258**	**0.1316**	**0.0405**	**10.7814**
User-based	0.0042	0.0305	0.0073	2.5770	0.0122	0.0753	0.0202	4.9332
Item-based	0.0106	0.0731	0.0182	4.9866	0.0093	0.0443	0.0144	3.2146
Dimensionality Reduction	0.0064	0.0408	0.0109	3.0120	0.0191	0.1026	0.0305	6.9486
Generative Model	0.0070	0.0466	0.0118	2.1084	0.0251	0.1273	0.0393	**11.0287**
Spreading Activation	0.0130	0.0863	0.0219	5.2209	0.0231	0.1155	0.0362	9.4955
Link Analysis	0.0133	0.0891	0.0224	6.4074	**0.0267**	0.1282	0.0415	10.3835
PRM	**0.0191**	**0.1343**	**0.0326**	**9.3014**	**0.0267**	**0.1354**	**0.0417**	**11.1411**

Table 5. Comparison with existing collaborative filtering algorithms (boldfaced measures were not significantly different from the largest measure at the 5% significance level).

5. Conclusions and Future Directions

In this paper we have presented a unified recommendation framework based on probabilistic relational models treating the recommendation problem as a special type of relational learning problem. We extended PRMs by introducing multi-set operators to explore the important relational data patterns relevant to recommendation. We also specialized the dependency structure searching and parameter estimation procedures to exploit the characteristics of the recommendation problem. The proposed PRM-based recommendation framework allows the integration of various existing recommendation approaches as well as exploitation of a wider range of relational data patterns. Our experimental results using data provided by an online bookstore and an online clothing merchant showed that such a unified framework resulted in improved recommendation performance over existing approaches.

We are in the process of extending our work in the following directions: (a) application of complex multi-set aggregation operations that provide richer input information for learning; (b) optimization of the algorithm implementation to improve space and time efficiencies; (c) a complete comparison with a wide range of existing hybrid recommendation approaches. Recommendation is only one special application of PRM-based relational learning. We can also develop other specializations of PRMs for a wide range of personalization, marketing, and managerial applications through modeling of consumer behavior using rich information available in large-scale retailing databases [17].

6. Acknowledgement

This research is partly supported by an NSF Digital Library Initiative-II grant, “High-performance Digital Library Systems: From Information Retrieval to Knowledge Management,” IIS-9817473, April 1999 – March 2002, and by an NSF Information Technology Research grant, “Developing a Collaborative Information and Knowledge Management Infrastructure,” IIS-0114011, September 2001 – August 2005.

7. References

1. J.S. Breese, D. Heckerman and C. Kadie, Empirical Analysis of Predictive Algorithms for Collaborative Filtering, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 43-52, (1998).
2. J. Cheng and R. Greiner, Comparing Bayesian Network Classifiers, in: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, 101-107, (1999).
3. M. Deshpande and G. Karypis, Item-Based Top-N Recommendation Algorithms, ACM Transactions on Information Systems, 22(1), 143-177, (2004).
4. S. Dzeroski and N. Lavrac, Relational Data Mining (Springer-Varlag, Berlin, 2001).
5. N. Friedman, L. Getoor, D. Koller and A. Pfeffer, Learning Probabilistic Relational Models, in: Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI), 1300-1307, (1999).

6. L. Getoor, N. Friedman, D. Koller and B. Taskar, Learning Probabilistic Models of Link Structure, Journal of Machine Learning Research, 3, 679-707, (2002).
7. L. Getoor and M. Sahami, Using Probabilistic Relational Models for Collaborative Filtering, in: Proceedings of the WebKDD'99, (1999).
8. K. Goldberg, T. Roeder, D. Gupta and C. Perkins, Eigentaste: A Constant Time Collaborative Filtering Algorithm, Information Retrieval, 4(2), 133-151, (2001).
9. J.L. Herlocker, J.A. Konstan, L.G. Terveen and J.T. Riedl, Evaluating Collaborative Filtering Recommender Systems, ACM Transactions on Information Systems, 22(1), 5-53, (2004).
10. T. Hofmann, Latent Semantic Models for Collaborative Filtering, ACM Transactions on Information Systems, 22(1), 89-115, (2004).
11. Z. Huang, H. Chen and D. Zeng, Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering, ACM Transactions on Information Systems (TOIS), 22(1), 116-142, (2004).
12. Z. Huang, W. Chung and H. Chen, A Graph Model for E-Commerce Recommender Systems, Journal of the American Society for Information Science and Technology (JASIST), 55(3), 259-274, (2004).
13. Z. Huang, D. Zeng and H. Chen, A Link Analysis Approach to Recommendation with Sparse Data, in: Proceedings of the Americas Conference on Information Systems, 1997-2005, (2004).
14. Z. Huang, D. Zeng and H. Chen, A Comparative Study of Recommendation Algorithms for E-Commerce Applications, IEEE Intelligent Systems, submitted, (2005).
15. D. Koller and A. Pfeffer, Probabilistic Frame-Based Systems, in: Proceedings of the Fifteenth Conference of the American Association for Artificial Intelligence, 580-587, (1998).
16. B.J. Mirza, B.J. Keller and N. Ramakrishnan, Studying Recommendation Algorithms by Graph Analysis, Journal of Intelligent Information Systems, 20(2), 131-160, (2003).
17. B.P.S. Murthi and S. Sarkar, The Role of the Management Sciences in Research on Personalization, Management Science, 49(10), 1344-1362, (2003).
18. J. Newton and R. Greiner, Hierarchical Probabilistic Relational Models for Collaborative Filtering, in: Proceedings of the ICML 2004 Workshop on Statistical Relational Learning and its Connections to Other Fields, (2004).
19. M. Pazzani, A Framework for Collaborative, Content-Based and Demographic Filtering, Artificial Intelligence Review, 13(5), 393-408, (1999).
20. C. Perlich and F. Provost, Aggregation-Based Feature Invention for Relational Learning, in: Proceedings of the Ninth SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003), (2003).
21. D. Poole, Probabilistic Horn Abduction and Bayesian Networks, Artificial Intelligence, 64, 81-129, (1993).
22. P. Resnick and H. Varian, Recommender Systems, Communications of the ACM, 40(3), 56-58, (1997).
23. B. Sarwar, G. Karypis, J. Konstan and J. Riedl, Application of Dimensionality Reduction in Recommender Systems: A Case Study, in: Proceedings of the WebKDD Workshop at the ACM SIGKKD, (2000).
24. B.M. Sarwar, G. Karypis, J.A. Konstan and J.T. Riedl, Item-Based Collaborative Filtering Recommendation Algorithms, in: Proceedings of the Tenth International World Wide Web Conference, 285-295, (2001).
25. J. Schafer, J. Konstan and J. Riedl, E-Commerce Recommendation Applications, Data Mining and Knowledge Discovery, 5(1-2), 115-153, (2001).
26. L.H. Ungar and D.P. Foster, A Formal Statistical Approach to Collaborative Filtering, in: Proceedings of the CONALD'98, (1998).