RC23735 (W0510-014) October 4, 2005
Computer Science

IBM Research Report

Relational Learning for Customer Relationship Management

Claudia Perlich
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Zan Huang
Smeal College of Buiness
The Pennsylvania State University
University Park, PA

Relational Learning for Customer Relationship Management

Claudia Perlich
Predictive Modeling Group
IBM T.J. Watson Research Center
Yorktown Heights, NY

Zan Huang
Supply Chain and Information Systems
Smeal College of Business
The Pennsylvania State University
University Park, PA

Abstract

Customer modeling is a critical component of customer relationship management (CRM). Successful customer modeling requires a holistic view and the consolidation of all customer information available to the business, which is typically stored in a relational database. With this understanding, customer modeling in CRM can be viewed as a special case of the *relational learning* problem, a recent extension of the traditional machine learning problem that aims to model the relational interdependencies within a database containing multiple interlinked tables. We establish in this paper the connection between relational learning and CRM analysis through detailed discussion of the tasks of customer classification and product recommendation, supported by examples of empirical results on seven real-world CRM data sets. We demonstrate that relational learning approaches can be valuable tools for a variety of CRM modeling tasks and discuss limitations and CRM specific extensions of these general relational learning approaches.

1. Introduction and Motivation

Customer relationship management (CRM), at a high level, can be viewed as the process of constructing a detailed database of customer information and interactions, modeling customer behaviors and preferences using such a database, and turning the predictions and insights into marketing actions to achieve the strategic goals of identifying, attracting, and retaining customers. Typical CRM modeling tasks include product recommendation, personalization, and the analysis of factors driving customer retention and loyalty. The underlying customer database stores all information that is available to the merchant about his customers. The CRM modeling practice and research can then be framed as a task of building explanatory or predictive customer models based on variables derived from this database. Examples of these variables typically include customers' demographics, purchase patterns reflected in sales transactions, linkage to products through sales transactions, linkage to other customers through overlapping purchased products, and others. While many of the CRM modeling tasks can be framed as probability estimation problems, the importance of the initial step of variable and feature construction cannot be overestimated. Traditional statistical modeling techniques like logistic regression and discriminant analysis provide the necessary ability of model estimation and inference, but assume a much simpler single-table representation as well as independence between observations. Aside from the customer demographics, all other relevant transaction and linkage information has to be manually transformed and condensed into descriptive variables on a customer level.

This class of modeling tasks based on a *multiple*- rather than *single-table representation* has recently received increased attention under the term of *relational learning* (Dzeroski et al. 2001). The major breakthrough that relational learning can bring to CRM is the *automation* of the process of constructing features from the secondary tables in the customer database and the feature selection process, which is currently performed more or less in a hand-crafting manner heavily relying on heuristics and domain expertise. The automated construction of features can provide new insights that improve the understanding of the customer preferences and behaviors and the effectiveness of marketing activity. The methods of relational learning are not limited to a single database and can be used across multiple data storage units as well as in distributed environments, as long as it is possible to match customers across the information sources.

The main objective of this paper is to establish the connection between the CRM modeling and the relational learning problem and to promote the development of customized relational learning approaches for CRM analysis. We are looking at two common classes of modeling tasks within CRM, 1) predictive customer modeling (in particular classification and probability estimation for cost-sensitive decision making) where the target (e.g., whether or not a customer will respond to a specific special offer) is known for a small set of customers and 2) product recommendation where we need to find the products that are of most interest to individual customers. Focusing on the automatic feature construction capability of relational learning approaches, we show how several of the traditional CRM models including the RFM (recency, frequency, monetary) and various recommendation approaches can be expressed within the general relational modeling framework. In addition to emulating these well-known approaches, relational learning can explore automatically a much larger set of potential models and find new and predictive dependencies that improve the model performance and provide new insights about customer behavior.

We provide in this paper several examples of relational models for customer classification and product recommendation tasks on seven real-world CRM domains. These examples clearly demonstrate the potential of the relational learning approaches for CRM applications. In addition, we discuss shortcomings of current relational learning approaches in relation to specific properties of CRM tasks and point out future research to address these limitations.

2. Relational Customer Databases and Relational Learning

Figure 1 shows a simple customer database from a book retailer, which contains three basic tables that store the information regarding customers, books, and sales transactions. We also include two additional tables containing keywords and their occurrence in the books, which enable the keyword searching capability. Typical customer databases are much more complex and may include additional tables that contain pre-purchase (e.g., Webpage browsing) and post-purchase (e.g., email communications) customer activities and customer responses to marketing programs.

Customer								
id	city	birth_year	education	vocation	sex	married	child	future_value
c1	new york	1977	college	financial	f	yes	1	high
c2	los angeles	1968	high school	construction	m	no	0	low
c3	seattle	1982	college	student	m	no	0	low

Book			
id	publisher	category	price
b1	p1	children	30
b2	p2	fiction	40
b3	p2	fiction	55
b4	p3	romance	25

Order		
customer	book	date
c1	b1	5/4/2001
c1	b2	6/1/2002
c2	b3	3/2/2001
c3	b2	7/12/2000
c3	b4	1/5/2001

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

Word	
id	word
w1	word1
w2	word2
w3	word3
w4	word4
w5	word5
w6	word6

Figure 1. An example book store customer database

Within a relational learning framework, a database not only serves for data storage and access but also forms the basis for building relational statistical models. We use this simple example to illustrate the relational learning feature space associated with such a database for customer modeling purposes.

We leverage the notation of probabilistic relational models (PRMs) (Getoor et al. 2002; Koller et al. 1998; Poole 1993) to facilitate our discussion. Using the example in Figure 1, we give some examples of the meaning of our notation. A relational database is formally represented by a relational schema R describing a set of tables $\boldsymbol{X}$. Each $X\in\boldsymbol{X}$ is associated with a set of *descriptive attributes* $\boldsymbol{A}(X)$ and a set of *reference slots* (e.g., foreign keys) $\boldsymbol{R}(X)$. We denote the attribute A of table X as $X.A$, which takes on a range of values $V(X.A)$. Customer.birth_year represents for instance the attribute birth_year of the Customer table and V(Customer.birth_year) is numeric. We denote the reference slot ρ of X as $X.\rho$, where ρ is associated with a one-to-one or many-to-one mapping from observations (rows) in table X to observations in another table Y with identical value of the identifier attribute $Y.id$. For convenience, we will assume that the slot name ρ is identical to the table name Y. For example Order.book is the reference slot book in the Order table that points to the corresponding observations in the Book table. Brackets [] represent the mapping operation and [Order.book] corresponds to the books in the Book table associated with an order. [Order.customer].education represents the education of a customer associated through the reference slot [Order.customer] with an order. For each reference slot ρ we define an inverse reference slot ρ^{-1} that represents the reverse (potentially one-to-many) mapping. [Order.customer]$^{-1}$ captures the mapping from a customer to the associated orders.

Using a chain of reference slots (including inverse ones) $\tau = \rho_1.\rho_2 \ldots .\rho_k$ we can define more complex relationships. Customer.[Order.customer]$^{-1}$.[Order.book].price represents the prices of the set of books bought by the customer. If any of the reference slots in the slot chain involves a one-to-many mapping, the derived attribute will be a multi-valued attribute. In this example [Order.customer]$^{-1}$ maps a customer to multiple orders, making Customer.[Order.customer]$^{-1}$.[Order.book].price a multi-valued attribute. Such multi-valued attributes require aggregation as discussed in more detail in Section 3.

Attributes like the ones introduced above form the inherent relational feature. In general, statistical models can be constructed to describe the dependency among all the potential attributes. The classification model is a simple form of such dependency models, where for each observation in the *target table* a particular *target attribute* is predicted using all other related attributes. For example, when classifying the customers into high and low future value types, Customer is the target table and Customer.future_value is the target attribute. All other related attributes including simple attributes such as Customer.vocation and complex attributes derived from reference slot chains such as *sum*{Customer.[Order.customer]$^{-1}$.[Order.book].price} (total past revenue from the customer) can enter the classification model as predictors. The recommendation model is another example, where the objective is to estimate the probability for a previously unobserved consumer-product pair to appear in the Order table (the likelihood that a customer will buy the book in the future). Relational attributes of various forms may contribute to the predictions, including the attributes of the associated customer, attributes of the associated products, and attributes jointly derived from the consumer-product pair.

Relational learning methods operate directly on such a feature space and substantially extend the capability of modeling different aspects of customer behavior and preference compared to traditional modeling techniques that only operate on feature vector representations of a single Customer table. Historically, relational learning was dominated by Inductive Logic Programming (ILP) which employs First-Order Logic clauses to build binary classification models. However, recent work has recognized the inherent uncertainty in many important application domains including CRM. Addressing this need, modern probabilistic relational approaches include the transformation of the originally relational domain into a traditional single-table representation (domain downgrading or propositionalization) and the upgrading of for instance Bayesian networks to represent multiple entity types and the dependencies between them. Modeling for CRM applications can profit significantly from this recent development in the relational learning field. The relational nature of the data is only one characteristic of CRM domains. Other important properties include the cost-sensitive nature of most marketing decisions, the need for model analysis and statistical inference, and finally the inherent uncertainty of human behavior. All these properties call for sophisticated probabilistic and decision theoretical modeling approaches that are now readily available. Either approach of domain downgrading or upgrading of Bayesian models depends critically on the expressiveness of feature construction.

3. Automated Feature Construction in Relational Learning Approaches

Current CRM models rely heavily on traditional data analysis methods and are operating on the same inherent feature space as discussed previously. However, significant domain knowledge and human judgment are involved during the process of deriving predictive features to be used by the customer models. For example, total past revenue from a customer is a basic measure used in CRM analysis to assess the customer's future value. This feature can be derived as Customer.[Order. customer]$^{-1}$.[Order.book].price from the multi-valued attribute. A relational learning approach automates this modeling process by constructing all potential features from the inherent feature spaces through generic feature construction mechanisms followed by a feature selection process to identify relevant features to be included into the models. The expressiveness of the resulting model heavily depends on the comprehensiveness of the feature construction mechanisms.

We describe in detail in this section several generic feature construction mechanisms that can replicate many features used in traditional CRM analysis as well as provide additional interesting new features that are potentially of great value for CRM analysis.

3.1 Simple Aggregation

An attribute derived from a slot chain $X.\tau.B$ will be multi-valued, if the reference chain τ consists of a reference slot ρ that corresponds to a one-to-many relationship. The values of these attributes can be numerical (e.g., the price attribute) or categorical (e.g., the category attribute). For a numerical multi-valued attribute, frequently used aggregation operators include *maximum, minimum, sum, median*, and *average*. For a categorical attribute, *mode* and *cardinality* are the meaningful aggregation operators. This simple collection of aggregators is, under certain assumptions on the maintenance of the database, sufficient to represent the traditional RFM model. Recency captures how recently a customer bough a product. This is captured by the last data in the order table *maximum*{Customer.[Order.customer]$^{-1}$.date}. For both the frequency and monetary value we have to assume that the current view of the database only includes a limited transaction history of a constant time period (e.g., one year). Under this assumption *sum*{Customer.[Order.customer]$^{-1}$.[Order.book].price} captures the monetary value and *cardinality*{Customer.[Order.customer]$^{-1}$} mirrors the frequency.

Another interesting and typically highly predictive feature for a classification modeling (e.g., the Customer.future_value as the target attribute) can be constructed from related instances in the target table and in particular their target attributes: Customer.[Order.customer]$^{-1}$.[Order.customer].future_value (future values of other customers who bought the same book(s) as the customer). Such features are the foundation of many network modeling approaches (e.g., Macskassy and Provost 2005) that have proven very effective in fraud detection and viral marketing (Domingos and Richardson 2002).

3.2. Distribution-Based Aggregation

The above mentioned simple aggregation operators are not suitable for attributes with many possible values and in particular the identifier attributes. Consider for instance the set of bought books Customer.[Order.customer]$^{-1}$.[Order.book].id. The *mode* operator is not well-defined since most products are bought only once and all identifiers will be unique. Even if the *mode* exists, such an aggregate loses almost all information and is unlikely to capture predictive information. In addition, the range of values of this new feature has excessively many possible values and renders it unsuitable for modeling.

The problem of modeling categorical attributes with many possible values is not new. A classical task of this nature is text classification based on the word occurrence. A simple and very effective approach is the Naïve Bayes classifier. It constructs the two class-conditional distributions over all words and makes predictions of a new document based on some distance metrics (e.g., likelihood, Euclidean, or cosine distance) between the document and the two class-conditional distributions. This simple mechanism can be seen as another form of aggregation of a multi-valued attribute. It is formally expressed as *cosine*{D_t, Customer.[Order.customer]$^{-1}$.[Order.book].id} where D_t is the target-conditional distribution that is estimated from the union of multi-value attributes

Customer.[Order.customer]$^{-1}$.[Order.book].id of all instances of customers for which the target attribute took the value *t* (e.g., Customer.future_value = high).

These distribution-based aggregates (Perlich and Provost 2005) extend the simple aggregation operators also in respect to its focus on *predictive* information. The particular value of a cosine distance will change as the target attribute values change. The values of the simple aggregates like *mean* and *mode* on the other hand will remain the same for a given multi-value attributes, independently of the particular classification task.

3.3 Set-Based Aggregation

A wide range of interesting attributes can only be constructed by aggregating multiple multi-valued attributes that use set operators such as *intersection* and *union*. The need for such set-based aggregation is most evident in modeling customers' product preferences for making recommendations. Customer.[Order.customer]$^{-1}$. [Order.book].[Order.book]$^{-1}$.[Order.customer].id represents the set of identifiers of customers who bought at least one common book as the target customer did (the customer neighbors) while Book.[Order.book]$^{-1}$.[Order.customer].id represents the set of identifiers of customers who bought the target book. These two multi-valued attributes, with aggregation operations, can provide certain information, such as the number of neighbors of the customer and the sales volume of the book, regarding the likelihood for the customer to purchase the book in the future. However, much more relevant information can be obtained by deriving the set similarity between the above two sets of customer identifiers through the cardinalities of the intersection and union of the two sets. Such information is essential for making recommendations and is closely related to a popular recommendation approach called *collaborative filtering* (Breese et al. 1998), which generates recommendation only using the transaction data based on the idea that customers with similar preferences revealed by the past transaction data will continue to behavior similarly in the future.

In fact all three major recommendation approaches including *content-based* (using the product attributes and transaction data), *demographic filtering* (using the customer attributes and transaction data), and *collaborative filtering* (using the transaction data only) (Huang et al. 2004a; Pazzani 1999; Resnick et al. 1997) can be generally emulated with relational features constructed by set-based aggregation. Similar to the attributes mentioned above, *cardinality*{*intersection*{Customer.[Order.customer]$^{-1}$. [Order.book].id, Book.[Occurrence.book]$^{-1}$.[Occurrence.word].[Occurrence.word]$^{-1}$. [Occurrence.book].id}} captures the essential information for *content-based* recommendation approaches by representing the content-based association between a customer-book pair: the number of books bought by the customer that contain words appearing in the book (i.e., the content similarity between the book and the customer's previously purchased books). Typical *demographic filtering* algorithms can also be emulated by such relational attributes involving customer attributes. For example, *cardinality*{*intersection*{Customer.birth_year, Book.[Order.book]$^{-1}$.[Order.customer].birth_year}} describes for a customer-book pair how many customers of similar age as the customer (the birth_year can be discretized into categorical values with the similar range of year of birth assigned same values) have purchased the book previously. Recommendations based on such attributes correspond to typical demographic filtering recommendations.

4. Examples of Empirical Work

The objective of this section is to illustrate the versatility of general relational learning techniques on a variety of CRM tasks. In particular, we do not intend to discuss in great detail the advantages and relative performances of different relational learning approaches or of relational learning vs. specialized marketing models, since we do not feel that we can do them justice. We rather compare the performance gain of applying a relational learner that takes advantage of additional information in additional tables beyond the naïve use of only customer demographics. We invested a minimum of effort in the domain preprocessing and used the relational learning algorithms with their standard parameter settings without optimizing the performance for a particular task.

4.1 Customer modeling tasks

We analyzed the applicability and performance of Automated Construction of Relational Attributes (ACORA) (Perlich and Provost 2005) as an example of a general relational learner that constructs simple and distribution-based features from all available sources of information including customer attributes, attributes of related entities, on a number of probability estimation and binary classification tasks for 6 different marketing domains:

- The **Sisyphus** data set was provided for a workshop at the 1998 PKDD conference. It is an excerpt from a data warehouse system of the private life insurance business at Swiss Life. The domain consists of 10 tables having between 500 and 100000 number of records. The Swiss Life Information Systems Research group provided two classification tasks, one of households and one for partners and we also tried to differentiate the customers by gender.

- The KDDCUP 2000 contains clickstream and purchase data from **Gazelle**.com, a legwear and legcare web retailer that closed their online store on 8/18/2000. We use the last month to construct the target and the data of the first 2 months for training and extract from the clickstream data regarding the content pages that a customer looked at. The domain has 4 tables with record numbers ranging from 3700 to 11142000. We build models for customer retention (Will a customer return in the last month?) and for customer loyalty (Will a customer buy something in the last month?)

- Blue Martini published, together with the data for the KDDCUP 2000, three additional customer datasets to evaluate the performance of association rule algorithms. We use the **BMS**-WebView-1 set of 59600 customers with a total of 146000 purchases in 497 distinct product categories. The objective is to identify customers who are most likely to purchase a product from one of three classes 12895, 110307, 110311 given all other items in the transaction (unfortunately no further information was provided about the nature of these classes).

- The **E-books** domain comprises data from a five-year-old Korean startup and contains two tables: a customer table with demographics and preferences and the transaction table (price, category, and identifier). The tasks include the estimation of 2 purchase probabilities for particular books and 3 customer demographics based on their purchasing behavior (gender, nation, and children). The domain has a total of 20000 customers and 544900 transactions.

- A **Banking** data set was provided by a Czech bank for the PKDD 99 Discovery Challenge. It contains a total of 8 tables of customer information with record numbers ranging from 77 to 1056300 including transactions, credit cards, demographics, loans, and accounts. No official task was suggested for the Challenge. We consider the following classification tasks: loan default, interest in credit card, and interest in life insurance.

- **ComScore** is a panelist-level database that captures detailed browsing and buying behavior of Internet users across the United States. The tasks were to identify 1) AMAZON customers and 2) customers that are open to cross selling (i.e., bought things other than books and music) while hiding the indication of visits to AMAZON.

Domain	**Task**	**Accuracy**			**AUC**	
		Prior	**Demo**	**Relational**	**Demo**	**Relational**
Sisyphus	Partner	0.5	0.6	**0.91**	0.68	**0.95**
	Household	0.82	0.82	**0.99**	0.53	**0.99**
	Gender	0.7	0.73	**0.81**	0.71	**0.85**
Gazelle	Buy	**0.98**	**0.98***	**0.98**	0.5*	**0.55**
	Return	**0.88**	**0.88***	**0.88**	0.5*	**0.59**
BMS	12895	0.94	0.94*	**0.97**	0.5*	**0.92**
	110307	0.94	0.94*	**0.98**	0.5*	**0.89**
	110311	0.95	0.95*	**0.97**	0.5*	**0.88**
E-Books	Common book	0.88	0.9	**0.94**	0.73	**0.99**
	Poetry	**0.97**	**0.97**	**0.97**	0.55	**0.77**
	Gender	0.51	0.72	**0.78**	0.75	**0.86**
	Nation	**0.96**	**0.96**	**0.96**	0.56	0.58
	Children	0.73	0.79	**0.83**	0.87	**0.91**
Banking	Loan Status	0.89	0.89	**0.93**	0.74	**0.93**
	Credit Card	**0.87**	**0.87**	**0.87**	**0.66**	**0.66**
	Insurance	**0.90**	**0.90**	**0.90**	0.49	**0.67**
ComScore	Amazon customer	0.84	0.843	**0.88**	0.55	**0.75**
	Amazon cross	0.64	0.635	**0.67**	0.55	**0.67**

Table 1: Accuracy and probability estimation (AUC) performances of relational learning approaches (Relational) in comparison to the propositional model using only the customer demographics (Demo, * if no demographics available). Boldfaced measures were not significantly different from the largest measure at the 5% significance level.

Table 1 shows the out of sample performance in terms of accuracy and area under the ROC (AUC) (Bradley 1997) of the relational learning system ACORA (Perlich and Provost 2005) using distances to the class-conditional distributions as well as standard aggregates for feature construction. We used logistic regression with feature selection as the model. The relative performances indicate that relational modeling almost always improved model performance: in 12 out of 18 tasks for accuracy and 17 out of 18 for probability estimation revealed by the AUC measure.

4.1 Product recommendation task

The **Book Store** dataset comprises data from a major Taiwan online bookstore. We focused on the five basic tables as shown in Figure 1: the *customer* table containing typical demographic information, the *book* table containing product attributes such as title, description, keywords, author, publisher, price, and number of pages, the *transaction* table containing customer and book identifiers as well as transaction time and other attributes like payment methods, and the *word* and *word occurrence* tables. We used this dataset to analyze the relational learner's capability in performing the recommendation task – producing a ranked list of *K* books for each customer as recommendation for future purchases based on the information provided by such a database. The data set we analyzed contained 3 years of transactions of a sample of 2,000 customers with a total of about 18,000 transactions and 9,700 books.

Using a unified recommendation framework based on the extension of a major relational learner, probabilistic relational models (PRMs), we were able to construct relational features based on the entire relational schema and automatically select the set of relevant features to build predictive models (Huang et al. 2004b). A PRM is an extension of *Bayesian networks* for describing probability distributions over a relational database. To model the recommendation problem, we added a special existence attribute (exist, with value of 1 representing observed transactions and 0 representing unobserved customer-book pairs as transactions) into the Order table and derive dependency models relevant to this Order.exist attribute. Our extension to the PRM modeling mainly involved the set-based aggregation introduced in Section 3.2.

Because the inherent feature space of the book dataset, different forms of relational attributes derived from the customer, product, and their interactions, the PRM resulting model emulates a hybrid recommendation approach. By restricting the feature space from which the predictive attributes are selected to collaborative features (attributes derived only from the Order table), content features (attributes derived from the Order, Book, and Occurrence tables) and demographic features (attributes derived from the Order and Customer tables) we also built models that emulate collaborative filtering, demographic filtering, content-based approaches. The performances of different recommendation approaches under this PRM-based recommendation framework (PRMR) are presented in Table 2 in comparison with the performance of a standard collaborative filtering algorithm based on customer neighborhood formation (Breese et al. 1998). Recommendation performance is measured by well-studied metrics including precision (probability of the recommended books to be actually purchased), recall (probability of books to be purchased being recommended), F-measure (harmonic mean of precision and recall) and rank score (which measures how well the correct recommendations are positioned in the ranked list). The PRM-based recommendation framework provided the basis for meaningful comparison across the three general recommendation approaches, as all aspects of the model construction and estimation for the different approaches were consistent only except for the restriction on the feature space.

Model (Algorithm)	**Precision**	**Recall**	**F-Measure**	**Rank Score**
Standard Collaborative	0.0122	0.0753	0.0202	4.9332
PRMR-Collaborative	0.0267	0.1354	0.0417	**11.1411**
PRMR-Content	0.0142	0.0767	0.0227	5.4225
PRMR-Demographic	0.0145	0.0778	0.0229	7.4946
PRMR-Hybrid	**0.0313**	**0.1636**	**0.0493**	**12.0511**

Table 2. Book Store dataset: Recommendation performance measures (*K*=10) of a standard collaborative filtering algorithm and the different models of a unified relational-learning-based recommendation framework (PRMR) that emulate various typical recommendation approaches (boldfaced measures were not significantly different from the largest measure at the 5% significance level).

We observe in Table 2 that the content-based and demographic filtering approaches had similar performances, the performance measures of the collaborative filtering approach almost doubled those of the content-based and demographic filtering approaches, and that the hybrid approach delivered the best performance with significant improvement compared to the collaborative filtering approach. All PRM-based models under different approaches outperformed the standard neighborhood-based collaborative filtering algorithm, which demonstrates the value of additional recommendation-relevant relational features constructed by the PRM-based recommendation framework that are not included in typical recommendation algorithms.

5. Limitations and Future Work on Relational Learning for CRM

Before we discuss more formally some of the limitations of general relational learning methods, let us consider a rather simple classification concept that cannot be expressed or learned with the methods we discussed earlier:

"Customers who bought increasingly more expensive goods"

Although the database contains all necessary information (price and time of purchases) to identify such customers, general relational learners cannot express or learn it. The reasons for this limitation are two essential assumptions that underlie the aggregation operations of almost all existing relational learning approaches. With the exception of count and set operations like union and intersection, standard aggregators like *mean* and *mode* only apply to sets of a single attribute, the price or the time. Such aggregators make two implicit assumptions that are violated by the above concept:

- Class-conditional independence between the attributes of related objects.
- Bags of related objects and their attributes are random samples.

The above concept expresses a fundamental interaction between price and time. The optimal feature that an aggregation operator should construct is the slope of price over time. Aggregating price and time separately, de facto destroys this relationship and the constructed features cannot be predictive. Dependencies of this sort are abundantly common in domains that include time or order. However, these assumptions are not just convenient simplifications. The more fundamental problem is the potentially huge search space of all possible dependencies in real world relational domains. Imposing assumptions of independence is one approach to limit the search space and improve the reliability of the resulting models.

The capability to express and learn temporal dependencies is an important step to improve the applicability of general relational learning techniques to CRM tasks. It would be of value to analyze what types of dependencies are typically relevant in CRM and to formalize new aggregation operators similar to our earlier suggestion of a slope operator that can capture relationships between two numeric attributes. In particular, there are substantial recent advances in temporal data mining that specifically focus on capturing temporal and sequential data patterns (Roddick and Spiliopoulou 2001). We expect to see valuable development by combining the temporal and relational modeling of a comprehensive evolving customer databases for CRM applications.

Finally, it is worthwhile to observe that despite the limitations of single-attribute aggregation and the inability to express the concept explicitly, existing relational models that aggregate identifiers may still be able to make prediction according to the true concept. Identifiers are not simply another set of categorical attributes. They represent implicitly the joint occurrence of all attributes including even unobserved ones. Consider the above example in the case of a vendor of top-notch electronic devices where inventory is changing over time. It is therefore possible to associate a noisy estimate of purchase time with each product. The positive class-conditional distribution over the product identifiers will show higher probabilities for older cheap and newer expensive products and low probabilities for recent cheap and old expensive product identifiers. This pattern may be sufficient to construct discriminative vector distances for the classification model. This mechanism of identifier aggregation can be highly effective even if independence assumptions are violated. It even enables modeling concepts of unobserved properties. It is not necessary to observe that “Harry Potter” and “Alice in Wonderland” are books for teenagers; the model will simply construct a class-conditional vector that has higher probabilities for such books. This also implies a major shortcoming of such models: It becomes increasingly difficult to understand why the model is performing well. One of the immediate tasks is to provide visualization tool that allow the analysis of the model components including the class-conditional distributions.

6. Summary and Conclusions

We have demonstrated that relational machine learning approaches can be valuable tools for a variety of modeling tasks for customer relationship management including customer classification/probability estimation and product recommendation. They can lift the heavy burden of domain exploration and feature construction from the shoulder of the domain experts and provide new and interesting insights about customer behavior that were not known before. These tools are not meant to fully take over CRM modeling but rather to provide initial support in the exploration of the domain and the search for relevant information. To achieve optimal performance, the currently employed feature construction methods need to be extended beyond the discussed operators to capture relevant concepts in human behavior that is linked to time. Additional work will be needed to make the learned models more accessible and interpretable to the domain expert by providing visualization and model analysis functionality. Another direction to make relational learning models attractive and effective is to provide user interfaces that allow the definition of prior knowledge about the dependencies of the domain.

References

Bradley, A.P. "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition* (30:7) 1997, pp. 1145-1159.

Breese, J.S., Heckerman, D., and Kadie, C. "Empirical analysis of predictive algorithms for collaborative filtering," Fourteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann., Madison, WI, 1998, pp. 43-52.

Domingos, P., Richardson, M. "Mining Knowledge-Sharing Sites for Viral Marketing," Eighth International Conference on Knowledge Discovery and Data Mining, Edmonton, Canada, 2002, pp. 61-70.

Dzeroski, S., and Lavrac, N. *Relational Data Mining* Springer-Varlag, Berlin, 2001.

Getoor, L., Friedman, N., Koller, D., and Taskar, B. "Learning probabilistic models of link structure," *Journal of Machine Learning Research* (3) 2002, pp. 679-707.

Huang, Z., Chung, W., and Chen, H. "A graph model for e-commerce recommender systems," *Journal of the American Society for Information Science and Technology* (55:3) 2004a, pp. 259-274.

Huang, Z., Zeng, D., and Chen, H. "A unified recommendation framework based on Probabilistic Relational Models," Fourteenth Annual Workshop on Information Technologies and Systems (WITS), Washington, DC, 2004b, pp. 8-13.

Koller, D., and Pfeffer, A. "Probabilistic frame-based systems," Fifteenth Conference of the American Association for Artificial Intelligence, Madison, Wisconsin, 1998, pp. 580-587.

Macskassy, S.A. and Provost F. "Suspicion scoring based on guilt-by-association, collective inference, and focused data access," In Proceedings of the NAACSOS Conference, Washington, DC, 2005.

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

Perlich, C., and Provost F., "ACORA: Distribution-Based Aggregation for Relational Learning from Identifier Attributes," Forthcoming in special issue on Statistical Relational Learning and Multi-Relational Data Mining, *Journal of Machine Learning*, 2005.

Poole, D. "Probabilistic Horn abduction and Bayesian networks," *Artificial Intelligence* (64) 1993, pp. 81-129.

Resnick, P., and Varian, H. "Recommender systems," *Communications of the ACM* (40:3) 1997, pp. 56-58.

Roddick, J. F. and Spiliopoulou, M. "A Survey of Temporal Knowledge Discovery Paradigms and Methods," *IEEE Transactions on Knowledge and Data Engineering* (14:4) 2001, pp. 750-767