Aggregation for Predictive Modeling with Relational Data

Claudia Perlich
IBM Research, USA

Foster Provost
New York University, USA

INTRODUCTION

Most data mining and modeling techniques have been developed for data represented as a single table, where every row is a feature vector that captures the characteristics of an observation. However, data in most domains are not of this form and consist of multiple tables with several types of entities. Such relational data are ubiquitous; both because of the large number of multi-table relational databases kept by businesses and government organizations, and because of the natural, linked nature of people, organizations, computers, and etc. Relational data pose new challenges for modeling and data mining, including the exploration of related entities and the aggregation of information from multi-sets ("bags") of related entities.

BACKGROUND

Relational learning differs from traditional feature-vector learning both in the complexity of the data representation and in the complexity of the models. The relational nature of a domain manifests itself in two ways: (1) entities are not limited to a single type, and (2) entities are related to other entities. Relational learning allows the incorporation of knowledge from entities in multiple tables, including relationships between objects of varying cardinality. Thus, in order to succeed, relational learners have to be able to identify related objects and to aggregate information from bags of related objects into a final prediction.

Traditionally, the analysis of relational data has involved the manual construction by a human expert of attributes (e.g., the number of purchases of a customer during the last three months) that together will form a feature vector. Automated analysis of relational data is becoming increasingly important as the number and complexity of databases increases. Early research on automated relational learning was dominated by Inductive Logic Programming (Muggleton, 1992), where the classification model is a set of first-order-logic clauses and the information aggregation is based on existential unification. More recent relational learning approaches include distance-based methods (Kirsten et al., 2001), propositionalization (Kramer et al., 2001; Knobbe et al., 2001; Krogel et al., 2003), and upgrades of propositional learners such as Naïve Bayes (Neville et al., 2003), Logistic Regression (Popescul et al., 2002), Decision Trees (Jensen & Neville, 2002) and Bayesian Networks (Koller & Pfeffer, 1998). Similar to manual feature construction, both upgrades and propositionalization use Boolean conditions and common aggregates like min, max, or sum to transform either explicitly (propositionalization) or implicitly (upgrades) the original relational domain into a traditional feature-vector representation.

Recent work by Knobbe et al. (2001) and Wrobel & Krogel (2001) recognizes the essential role of aggregation in all relational modeling and focuses specifically on the effect of aggregation choices and parameters. Wrobel & Krogel (2003) present one of the few empirical comparisons of aggregation in propositionalization approaches (however with inconclusive results). Perlich & Provost (2003) show that the choice of aggregation operator can have a much stronger impact on the resultant model's generalization performance than the choice of the model induction method (decision trees or logistic regression, in their study).

MAIN THRUST

For illustration, imagine a direct marketing task where the objective is to identify customers who would respond to a special offer. Available are demographic information and all previous purchase transactions, which include PRODUCT, TYPE and PRICE. In order to take advantage of these transactions, information has to be aggregated. The choice of the aggregation operator is crucial, since aggregation invariably involves loss of (potentially discriminative) information.

Typical aggregation operators like min, max and sum can only be applied to sets of numeric values, not to

objects (an exception being count). It is therefore necessary to assume class-conditional independence and aggregate the attributes independently, which limits the expressive power of the model. Perlich & Provost (2003) discuss in detail the implications of various assumptions and aggregation choices on the expressive power of resulting classification models. For example, customers who buy mostly expensive books cannot be identified if price and type are aggregated separately. In contrast, ILP methods do not assume independence and can express an expensive book (TYPE="BOOK" and PRICE>20); however aggregation through existential unification can only capture whether a customer bought at least one expensive book, not whether he has bought primarily expensive books. Only two systems, POLKA (Knobbe et al., 2001) and REGLAGGS (Wrobel & Krogel, 2001) combine Boolean conditions and numeric aggregates to increase the expressive power of the model.

Another challenge is posed by categorical attributes with many possible values, such as ISBN numbers of books. Categorical attributes are commonly aggregated using mode (the most common value) or the count for all values if the number of different values is small. These approaches would be ineffective for ISBN: it has many possible values and the mode is not meaningful since customers usually buy only one copy of each book. Many relational domains include categorical attributes of this type. One common class of such domains involves networked data, where most of the information is captured by the relationships between objects, possibly without any further attributes. The identity of an entity (e.g., Bill Gates) in social, scientific, and economic networks may play a much more important role than any of its attributes (e.g., age or gender). Identifiers such as name, ISBN, or SSN are categorical attributes with excessively many possible values that cannot be accounted for by either mode or count.

Perlich and Provost (2003) present a new multi-step aggregation methodology based on class-conditional distributions that shows promising performance on networked data with identifier attributes. As Knobbe et al. (1999) point out, traditional aggregation operators like min, max, and count are based on histograms. A histogram itself is a crude approximation of the underlying distribution. Rather than estimating one distribution for every bag of attributes, as done by traditional aggregation operators, this new aggregation approach estimates in a first step only one distribution for each class, by combining all bags of objects for the same class. The combination of bags of related objects results in much better estimates of the distribution, since it uses many more observations. The number of parameters differs across distributions: for a normal distribution only two parameters are required, mean and variance, whereas distributions of categorical attributes have as many parameters as possible attribute values. In a second step, the bags of attributes of related objects are aggregated through vector distances (e.g., Euclidean, Cosine, Likelihood) between a normalized vector-representation of the bag and the two class-conditional distributions.

Imagine the following example of a document classification domain with two tables (Document and Author) shown in *Figure 1*.

The **first aggregation step** estimates the class-conditional distributions $D_{Class\ n}$ of authors from the Author table. Under the alphabetical ordering of position:value pairs, 1:A, 2:B, and 3:C, the value for $D_{Class\ n}$ at position k is defined as:

$$D_{Class\ n}[k] = \frac{\text{Number of occurrences of author k in the set of authors related to documents of class n}}{\text{Number of authors related to documents of class n}}$$

The resulting estimates of the class-conditional distributions for our example are given by:

$$D_{Class\ 0} = [0.5\ \ 0\ \ 0.5] \text{ and } D_{Class\ 1} = [0.4\ \ 0.4\ \ 0.2]$$

The **second aggregation step** is the representation of every document as a vector:

$$D_{Pn}[k] = \frac{\text{Number of occurrences of author k related to the document Pn}}{\text{Number of authors related to document Pn}}$$

The vector-representation for the above examples are $D_{P1} = [1\ 0\ 0]$, $D_{P2} = [0.5\ 0.5\ 0]$, $D_{P3} = [0.33\ 0.33\ 0.33]$, and $D_{P4} = [0\ 0\ 1]$.

The **third aggregation step** calculates vector distances (e.g., cosine) between the class-conditional distribution and the documents $D_{P1},...,D_{P4}$. The new Document table with the additional cosine features is shown in *Figure 2*. In this simple example, the distance from $D_{Class\ 1}$ separates the examples perfectly; the distance from $D_{Class\ 0}$ does not.

Figure 1. Example domain with two tables that are linked through Paper ID

Document Table	
Paper ID	**Class**
P1	0
P2	1
P3	1
P4	0

Author Table	
Paper ID	**Author Name**
P1	A
P2	B
P2	A
P3	B
P3	A
P3	C
P4	C