RC23859 (W0601-169) January 31, 2006
Computer Science

IBM Research Report

Learning from Identifier Attributes: Distribution-Based Aggregation for Relational Learning

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

Foster Provost
Department of Information, Organizations and Management Science
Stern School of Business
New York University
44 West 4th Street
New York, NY

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Learning from Identifier Attributes: Distribution-Based Aggregation for Relational Learning

Claudia Perlich[1], Foster Provost[2]

[1] IBM T.J. Watson Research Center
1101 Kitchawan Road, RTE 134/PO BOX 218
Yorktown Heights, USA
cperlich@stern.nyu.edu

[2] Department of Information, Organizations and Management Science
Stern School of Business, New York University
44 West 4th Street, New York, USA
fprovost@stern.nyu.edu

Abstract. Feature construction through aggregation plays an essential role in modeling relational domains with one-to-many relationships between tables. One-to-many relationships lead to bags (multisets) of related entities, from which predictive information must be captured. This paper focuses on aggregation from categorical attributes that can take many values (e.g., object identifiers). We present a novel aggregation method as part of a relational learning system ACORA, that combines the use of vector distance and meta-data about the class-conditional distributions of attribute values. We provide a theoretical foundation for this approach deriving a "relational fixed-effect" model within a Bayesian framework, and discuss the implications of identifier aggregation on the expressive power of the induced model. One advantage of using identifier attributes is the circumvention of limitations caused either by missing/unobserved object properties or by independence assumptions. Finally, we show empirically that the novel aggregators can generalize in the presence of identifier (and other high-dimensional) attributes, and also explore the limitations of the applicability of the methods.

1 Introduction

When building statistical models from relational data, modelers face several challenges. This paper focuses on one: incorporating one-to-many relationships between a target entity and related entities. In order for standard statistical modeling techniques to be applicable, bags (multisets) of related entities must be *aggregated*, where aggregation is the process of converting a bag of entities into a single value.

This paper addresses classification and the estimation of class-membership probabilities, and unless otherwise specified we will assume binary classification. We assume that data are represented as a multi-table relational database, although the techniques apply as well to other relational representations. The simplest domain that exhibits the one-to-many relationships at issue consists of two tables: a *target* table, which contains one row for each entity to be classified, and an auxiliary table that contains multiple rows of additional information about each target entity. Figure 1 illustrates the case of a customer table and a transaction table. This simple case is ubiquitous in business applications: for customer classification for churn management, direct marketing, fraud detection etc., it is important to consider transaction information (such as types, amounts, times, locations).

Customer

CID	CLASS
C1	0
C2	1
C3	1
C4	0

Transaction

CID	TYPE	ISBN	Price
C1	Fiction	523	9.49
C2	Non-Ficiton	231	12.99
C2	Non-Fiction	523	9.49
C2	Fiction	856	4.99
C3	Non-Fiction	231	12.99
C4	Fiction	673	7.99
C4	Fiction	475	10.49
C4	Ficiton	856	4.99
C4	Non-Fiction	937	8.99

Fig. 1. Example of a relational classification task consisting of a target table Customer(CID, CLASS) and a one-to-many relationship to the table Transaction(TID, TYPE, ISBN, Price).

For modeling with data such as these, practitioners traditionally have manually constructed presumably relevant features before applying a conventional ("propositional") modeling technique such as logistic regression. One group of automatic relational modeling approaches follows a similar process, explicitly constructing features from secondary tables in order to allow the application of standard statistical modeling techniques. The potential advantages of such a transformation, or "propositionalization," approach have been discussed previously [1].

Aggregation of bags of values plays an important role in the transformation process, but only two types of automated aggregation are regularly used: (1) simple aggregates, such as the arithmetic average or the most frequent value, and (2) domain-specific aggregates, such as recency and frequency of prior purchases (used regularly for direct marketing). These simple aggregates may be suitable for bags of numeric attributes or low-cardinality categoricals. However, applying them to high-dimensional categorical attributes results either in massive loss of information or an extremely large and sparse feature space—neither of which facilitates subsequent modeling.

We introduce and analyze novel aggregation methods[1] that are more complex than the simple aggregates, are general enough to construct features for a variety of modeling domains, and are tailored to statistical relational learning [3]. The main idea of the feature construction techniques is first to estimate and store (class-conditional) data about the distributions of the bags of attribute values (*distributional meta-data*). Second, when confronted with a particular target case we consider various vector distances to compress the information from the case's bag(s) relative to the distributional meta-data.

The main contributions of this work are:

1. It provides an initial theoretical and rhetorical analysis of principles for developing new aggregation operators.
2. It develops a novel method for relational feature construction, based on this analysis, including novel aggregation operators. To our knowledge, this is the first relational aggregation approach that can be applied generally to categorical attributes with high cardinality.
3. It draws an analogy to the statistical distinction between random- and fixed-effect modeling, provides a theoretical justification for using class-conditional aggregation, and argues that it has the potential to overcome some of the shortcomings of traditional aggregates.
4. It provides an analysis of the aggregation of object identifiers, a common class of categorical attributes with high cardinality, which in a relational setting can provide important discriminative information. A unique opportunity presented by the introduction of object identifiers is that modeling may be able to learn from unobserved object properties.

Section 2 presents a general analysis of potential aggregation objectives for modeling, and derives the principles for developing aggregates. Section 3 presents our new aggregation operators and gives an overview of the relational learning system ACORA. Next we provide in Section 4 a brief analysis of the new aggregates and discuss the implications of applying such aggregations to object identifiers. Section 5 provides empirical support for our claims of the applicability of the novel aggregates to domains with high-dimensional categorical attributes and of the advantages of learning from object identifiers.

2 Setup and Principles of Aggregation

A relational probability estimation (or classification) task is defined by a relational database RDB containing two or more tables T_i, including a particular **target table** T_t. Every table T_i contains s_i rows of **instances** $t_i^f, (1 \leq f \leq s_i)$. Each instance f is represented by a set of n attribute values $t_i^f = (t_{i1}^f, \ldots, t_{in}^f)$. The **type**, $D(T_{im})$, of attribute T_{im} is either $\mathbb{R}$ in the case of numeric attributes, or the set of values that a categorical attribute T_{im} can assume; in cases where this is not known a priori, we define $D(T_{im}) = \bigcup_{f=1}^{k}(t_{im}^f)$, the set of values that

[1] This paper is an extension of the second half of [2].

are observed across all instances f in column m of table T_i. The **cardinality** of a categorical attribute $C(T_{im})$ is equal to the number of distinct values that the attribute can take: $C(T_{im}) = |D(T_{im})|$.

One particular attribute T_{tc} in the target table T_t is the **target**, a class label for which a model has to be learned given all the information in RDB. We will consider binary classification ($D(T_{tc}) = \{0, 1\}$). Assuming a cost function $\mathcal{C}(c, \hat{c})$ where c is a vector of binary class labels and $\hat{c}$ a vector of probabilities of class membership, we define the relational probability estimation task as finding a mapping $M* : (T_t^f, RDB) \rightarrow [0, 1]$ from instances in T_t (along with all information in RDB), subject to minimizing the cost in expectation over any possible set of target cases, T_t^*:

$$M* = \operatorname*{argmin}_M E_{T_t^*}[C(T_{tc}^*, M(T_t^*, RDB))]. \tag{1}$$

The main distinction between relational and propositional model estimation is the additional information in tables of RDB other than T_t. This additional information can be associated with instances in the target table via **keys**. The conventional definition of a key requires a categorical attribute T_{mj} to be unique across all rows in table T_m: $C(T_{mj}) = s_m$. A link to information in another table T_n is established if that key T_{mj} also appears as T_{nl} in another table T_n, where it would be called a **foreign key**. This definition of a foreign key requires an equality relational ER between the types of pairs of attributes $ER(D(T_{mj}), D(T_{nl}))$. We will assume that for the categorical attributes in RDB this equality relation is provided.

More fundamentally, keys are used to express semantic links between the real entities that are modeled in the RDB. In order to capture these links, in addition to entities' attributes we also must record an **identifier** that stands for the **true identity** of each entity. Although database keys often are true identifiers (e.g., social security numbers), all identifiers are not necessarily keys in a particular RDB. This can be caused either by a lack of normalization of the database or by certain information not being stored in the database. For example consider domains where no information is provided for an entity beyond a "name": shortnames of people in chatrooms, names of people transcribed from captured telephone conversations, email addresses of contributors in news groups. In such cases RDB may have a table to capture the relations between entities, but not a table for the properties of the entity. This would violate the formal definition of key, since there is no table where such an identifier is unique. An example of an identifier that is not a key is the ISBN attribute in the transaction table in Figure 1.

Without semantic information about the particular domain it is impossible to say whether a particular attribute reflects the identity of some real entity. A heuristic definition of identifiers can be based on the cardinality of its type (or an identical type under ER):

Definition 1: T_{mj} is an **identifier** if $D(T_{mj}) \neq \mathbb{R}$ and
$(\exists\, T_{gh} | C(T_{gh}) \geq I_{MIN}$ and $ER(D(T_{gh}), D(T_{mj})))$.

Informally, a identifier is a categorical attribute where the cardinality of its type or some equivalent type is larger than some constant I_{MIN}. Note that for many domains the distinction between keys and identifiers will be irrelevant because both definitions are true for the same set of attributes. If I_{MIN} is set to the size of the smallest table, the keys will be a subset of the identifiers. The use of identifiers to link objects in a database (still assuming an equality relation between pairs of attributes) will therefore provide at least as much information or more than the use of keys. The choice of I_{MIN} is bounded from above by s_t, the size of the target table. There is no clear lower limit, but very small choices (e.g., below 50) for I_{MIN} are likely to have a detrimental effect on model estimation, in terms of run time and potentially also in terms of accuracy, because too many irrelevant features will be constructed.

A *relationship* between entities is defined by a quadruple of the form $(T_t, T_{tm}, T_n, T_{nq})$ consisting of the two tables and two identifiers of equivalent type. The bag R of objects related to t_t^f under this relationship is defined as $R_{T_n}(t_t^f) = \{t_n^y | t_{tm}^f = t_{nq}^y\}$ and the bag of related values of attribute T_{nz} is defined as $R_{T_{nz}}(t_t^f) = \{t_{nz}^y | t_{tm}^f = t_{nq}^y\}$.

Beyond defining related objects, identifiers are also important for aggregation. Aggregation operators are needed to incorporate information from one-to-many relationships as in our example in Figure 1, joining on CID. The challenge in this context is the aggregation of the ISBN attribute, which we assume has cardinality larger than I_{MIN}. An aggregation operator A provides a mapping from a bag of values $R_{T_{nz}}(t_t^f)$ to either $\mathbb{R}$, $(A : R_{T_{nz}}(t_t^f) \to \mathbb{R})$, or to the original type of the attribute $(A : R_{T_{nz}}(t_t^f) \to D(T_{nz}))$. Simple aggregation operators for bags of categorical attributes are the $COUNT$, value counts for all possible values $v \in D(T_{nz})$, and the $MODE$. $COUNT = |R_{T_{nz}}(t_t^f)|$ is the size of the bag. $COUNT_v$ for a particular value v is the number of times value v appeared in the bag $R_{T_{nz}}(t_t^f)$,

$$COUNT_v(R_{T_{nz}}(t_t^f)) = \sum_{e \in R_{T_{nz}} | e = v} 1. \tag{2}$$

The $MODE$ is the value v that appears most often in $R_{T_{nz}}(t_t^f)$:

$$MODE(R_{T_{nz}}(t_t^f)) = \underset{v}{\operatorname{argmax}} \, COUNT_v(R_{T_{nz}}(t_t^f)). \tag{3}$$

In the example, $MODE(R_{TYPE}(C2, 1))$ = 'Non-Fiction' for the bag of TYPE attributes related to customer C2. None of these simple aggregates is appropriate for high-cardinality attributes. For example, since most customers buy a book only once, for bags of ISBNs there will be no well-defined $MODE$. The number of counts (all equal to either zero or one) would equal the cardinality of the identifier's domain, and could exceed the number of training examples by orders of magnitude—leading to overfitting.

More generally and independently of our definition of identifiers, any categorical attribute with high cardinality poses a problem for aggregation. This has been recognized implicitly in prior work (see Section 6), but rarely addressed explicitly. Some relational learning systems [4] only consider attributes with cardinality of less than n, typically below 50; Woznica et al. [5] define *standard attributes* excluding keys, and many ILP systems require the explicit identification of categorical values to be considered for equality tests, leaving the selection to the user.

2.1 Design Principles for Aggregation Operators

Before we derive formally in Section 3.1 a new aggregation approach for categorical attributes with high cardinality, let us explore the objectives and potential guidelines for the development of aggregation operators.[2] The objective of aggregation in relational modeling is to provide features that improve the generalization performance of the model (the ideal feature would discriminate perfectly between the cases of the two classes). However, feature construction through aggregation typically occurs in an early stage of modeling, or one removed from the estimation of generalization performance (e.g., while following a chain of relations). In addition, aggregation almost always involves the loss of information. Therefore an immediate concern is to limit the loss of predictive information, or the general loss of information if predictive information cannot yet be identified. For instance, one measure of the amount of information loss is the number of aggregate values relative to the number of unique bags. For example for the variable TYPE in our example, there are 54 non-empty bags with size less than 10 containing values from {Fiction,Non-Fiction}. The $MODE$ has 2 possible aggregate values and $COUNT$ has 9. One could argue that the general information loss is larger in the case of $MODE$. In order to limit the loss and to preserve the ability to discriminate classes later in the process, it desirable to preserve the ability to discriminate *instances*:

Principle 1: Aggregations should capture information that discriminates instances.

Although instance discriminability is desirable, it is not sufficient for predictive modeling. It is simple to devise aggregators that involve no apparent information loss. For the prior example, consider the enumeration of all possible 54 bags or a prime-coding 'Non-Fiction'=2, 'Fiction'=3, where the aggregate value corresponding to a bag is the product of the primes. A coding approach can be used to express any one-to-many relationship in a simple feature-vector representation. An arbitrary coding would not be a good choice for predictive modeling, because it almost surely would obscure the natural similarity between bags: a bag with 5 'Fiction' and 4 'Non-Fiction' will be just as similar to a bag of 9 'Fiction' books as to a bag of 5 'Fiction' and 5 'Non-Fiction' books. In order for aggregation to produce useful features it must be aligned with the implicitly

[2] Related issues of quantifying the goodness of transformation operators have been raised in the context of "good kernels" for structured data (Gaertner et al. [6]).

induced notion of similarity that the modeling procedure will (try to) take advantage of. In particular, capturing *predictive* information requires not just any similarity, but similarity with respect to the learning task given the typically Euclidean modeling space. For example, an ideal predictive numeric feature would have values with small absolute differences for target cases of the same class and values with large absolute differences for objects in different classes. This implies, that the aggregates should not be independent of the modeling task; if the class labels were to change, the constructed features should change as well.

Principle 2: Aggregates should induce a similarity with respect to the learning task, that facilitates discrimination by grouping together target cases of the same class.

Thus, we face a tradeoff between instance discriminability and similarity preservation. Coding maintains instance discriminability perfectly, but obscures similarity (without luck or some additional mechanism). $COUNT$ and $MODE$ on the other hand lose much instance discriminability, but will assign identical values to bags that are in some sense similar—either to bags of identical size, or to bags that contain mostly the same element. Again, because aggregation often is distant in the induction process from the assessment of the final objective, it may be difficult to select one appropriate similarity measure. Furthermore, since most similarity-preserving operators involve information loss, it might be advantageous to use multiple operators that, when combined, capture more information.

Principle 3: Various aggregations should be considered, reflecting different notions of similarity.

For our example, consider the following alternative aggregation. Rather than capturing all information into a single aggregate, construct 2 attributes, one count for each value 'Fiction' and 'Non-Fiction'. The two counts together maintain the full information. Unfortunately, constructing counts for all possible values is possible only if the number of values is small compared to the number of training examples.[3]

The design principles suggest particular strategies and tactics for aggregation:

- Directly use target (class) values to derive aggregates that already reflect similarity with respect to the modeling task.
- Use multiple aggregates to capture different notions of similarity.
- Use numeric aggregates, since they can better trade off instance discriminability and similarity.

[3] Model induction methods suitable for high dimensional input spaces may confer an advantage for such cases, as they often do for text. However, we will see later, for relational problems even producing single-number aggregations can lead to a large number of features.

We present in the next section a novel aggregation approach based on these principles, that is particularly appropriate for high-dimensional categorical variables.

3 Aggregation for Relational Learning

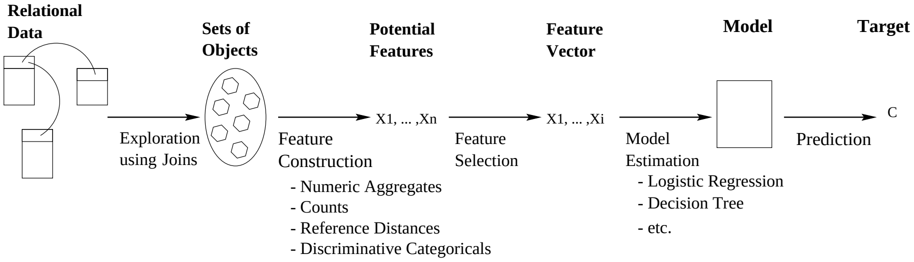

Fig. 2. System architecture of ACORA with four transformation steps: exploration, feature construction, feature selection, model estimation, and prediction. The first two (exploration and feature construction) transform the originally relational task (multiple tables with one-to-many relationships) into a corresponding propositional task (feature-vector representation).

In order to perform a comprehensive analysis of aggregation-based attribute construction for relational data, it is necessary to instantiate the ideas described above in a system that can be applied to non-trivial relational domains. ACORA (Automated Construction of Relational Attributes) is a propositionalization-based system that converts a relational domain into a feature-vector representation using aggregation to construct attributes automatically. ACORA consists of four nearly independent modules, as shown in Figure 2:

- exploration: constructing bags of related entities using joins and breadth-first search,
- aggregation: transforming bags of objects into single-valued features,
- feature selection, and
- model estimation.

Figure 3 outlines the ACORA algorithm more formally in pseudocode. Since the focus of this work is on aggregation we will concentrate in Section 3.1 on distribution-based aggregation assuming bags of values. The computation of such bags of related objects is explained in more detail in Section 3.2; it requires the construction of a domain graph where the nodes represent the tables the edges capture links between tables through identifiers. Following the aggregation, a feature selection procedure identifies valuable features for the modeling task, and in the final step ACORA estimates a classification model and makes predictions. Feature selection, model estimation, and prediction use conventional approaches including logistic regression, the decision tree learner C4.5 [7], and naive Bayes (using the WEKA package [8]), and are not discussed further in this paper.

ACORA Algorithm

Input: The domain specification (tables, attributes, types, and an equality relation over types), and a database RDB including a target table T_t with labeled training objects t_t and unlabeled test cases.

```
 1. Read specification and build domain graph $G$                 (Section 3.2)
 2. Initialize breadth-first list $L$ with target table: $L = \{T_t\}$
 3. Initialize feature table $F$ =non-identifier attributes($T_t$)
 4. Loop
 5.     $T_c$ = First($L$)
 6.     Foreach table $T_g$ in $RDB$ linked to $T_c$ in $G$ through some identifiers $T_{ck}, T_{gj}$
 7.         $J$ = Join $T_c$ and $T_g$ under the condition $T_{ck}=T_{gj}$
 8.         Foreach attribute $T_{ga}, a \neq j$                         (Section 3.1)
 9.             Foreach target observation $t_t^h$
10.                 Foreach applicable aggregation operator $A_s$
11.                     Construct $A_s(R_{T_{ga}}(t_t^f))$
12.                 End Foreach
13.                 Append aggregates $A_*$ as new columns to feature table $F$
14.             End Foreach
15.             Append to list $L$ the join result ($J$)
16.         End Foreach
17.         if (stopping criterion) GOTO Select Features
18.     End Foreach
19. End Loop
20. Select Features $SF$ from $F$
21. Build propositional model from $SF$
```

Fig. 3. Pseudocode of the ACORA algorithm

3.1 Aggregation using Distributional Meta-Data

The result of the join (on CID) of the two tables in our example database (step 7 in the pseudocode) is presented in Table 1. Consider the bag $R(C2, 1)$ of related transactions for customer C2:

$\langle$(C2,Non-Fiction,231,12.99),(C2,Non-Fiction,523,9.49), (C2,Fiction,856,4.99)$\rangle$.

CID	CLASS	TYPE	ISBN	Price
C1	0	Fiction	523	9.49
C2	1	Non-Fiction	231	12.99
C2	1	Non-Fiction	523	9.49
C2	1	Fiction	856	4.99
C3	1	Non-Fiction	231	12.99
C4	0	Fiction	673	7.99
C4	0	Fiction	475	10.49
C4	0	Fiction	856	4.99
C4	0	Non-Fiction	937	8.99

Table 1. Result of the join of the Customer and Transaction tables on CID for the example classification task in Figure 1. For each target case (C1 to C4) the one-to-many relationship can result in multiple entries (e.g., three for C2 and four for C4) highlighting the necessity of aggregation.

The objective of an aggregation operator A is to convert such a bag of related entities into a single value. In step 8 of the pseudocode, this bag of feature vectors is split by attribute into three bags $R_{TYPE}(C2,1) = \langle$Non-Fiction,Non-Fiction,Fiction$\rangle$, $R_{ISBN}(C2,1) = \langle$231,523,856$\rangle$, and $R_{Price}(C2,1) = \langle$12.99, 9.49,4.99$\rangle$. Aggregating each bag of attributes separately brings into play an assumption of class-conditional independence between attributes of related entities (discussed in Section 4.1 and [2]). ACORA may apply one or more aggregation operators to each bag. Simple operators that are applicable to bags of numeric attributes $R_{T_{ji}}$, such as Price, include the $SUM = \sum c \in R_{T_{ji}}$ or the $MEAN = SUM/|R_{T_{ji}}|$. Consider on the other hand $R_{ISBN}(C2,1) = \langle$231,523,856$\rangle$. ISBN is an example of a bag of values of an attribute with high cardinality, where the $MODE$ is not meaningful because it does not contain a "most common" element. The high cardinality also prevents the construction of counts for each value, because counts would result in a very sparse feature vector with a length equal to the cardinality of the attributes (often much larger than the number of training examples), which would be unsuitable for model induction.

Distances to Reference Vectors and Distributions The motivation for the new aggregation operators presented in the sequel is twofold: 1) to deal with bags of categorical high-cardinality attributes for which no satisfactory aggregation operators are available, and 2) to develop aggregation operators that satisfy the principles outlined in Section 2.1 in order ultimately to improve predictive performance. Note that even if applicable, the simple aggregates do not satisfy all the principles.

The main idea is to collapse the cardinality of the attribute by applying a vector distance to a vector representation both of the bag of related values and of a *reference bag* that is constructed across multiple related bags. In particular, the reference bags can be conditioned on the class label as follows. Let us define $B^f_{T_{ij}}$ as the *vector representation* of a bag of categorical values $R_{T_{ji}}(t^f_t)$. Specifically, given an ordering, $O : D(T_{ji}) \rightarrow \mathbb{N}$, and a particular value v of attribute T_{ji}, the value of $B^f_{T_{ij}}$ at position $O(v)$ is equal to the number of occurrences c_v of value v in the bag.

$$B^f_{T_{ij}}[O(v)] = c_v \tag{4}$$

For example $B^{C2}_{TYPE} = [2, 1]$ for $R_{TYPE}(C2, 1) = \langle$Non-Fiction,Non-Fiction,Fiction$\rangle$, under the order O(Non-Fiction)=1, O(Fiction)=2. We will use the term *case vector* to mean this vector representation of the bag of values related to a particular case.

Based on the case vectors in the training data, the algorithm constructs two class-conditional *reference vectors* B^0 and B^1 and an unconditional reference vector B^*, where s_1 is the number of positive target cases and s^0 is the number of negative target cases:

$$B^0_{T_{ij}}[O(v)] = \frac{1}{s_0} \sum_{f|t^f_{tc}=0} B^f_{T_{ij}}[O(v)] \tag{5}$$

$$B^1_{T_{ij}}[O(v)] = \frac{1}{s_1} \sum_{f|t^f_{tc}=1} B^f_{T_{ij}}[O(v)] \tag{6}$$

$$B^*_{T_{ij}}[O(v)] = \frac{1}{s_1 + s_0} \sum_{f} B^f_{T_{ij}}[O(v)] \tag{7}$$

$B^1_{T_{ij}}[O(V)]$ is the average number of occurrences of value v related to a positive target case ($t_{tc} = 1$) and $B^0_{T_{ij}}[O(v)]$ the average number of occurrences of a values v related to a negative target case ($t_{tc} = 0$). $B^*_{T_{ij}}[O(v)]$ is the average number of occurrences of values related to any target case. We also consider the following **normalized** versions D^0, D^1 and D^* that approximate the class-conditional and unconditional distributions from which the data would have been drawn, where r_f is the number of values related to t^f_t (the size of bag $R_{T_{ij}}(t^f_t)$) :

$$D^0_{T_{ij}}[O(v)] = \frac{1}{\sum_{f|t^f_{tc}=0} r_f} \sum_{f|t^f_{tc}=0} B^f_{T_{ij}}[O(v)] \tag{8}$$

$$D^1_{T_{ij}}[O(v)] = \frac{1}{\sum_{f|t^f_{tc}=1} r_f} \sum_{f|t^f_{tc}=1} B^f_{T_{ij}}[O(v)] \tag{9}$$

$$D^*_{T_{ij}}[O(v)] = \frac{1}{\sum_f r_f} \sum_{f} B^f_{T_{ij}}[O(v)] \tag{10}$$

For the example, the case vectors for TYPE and ISBN are shown in Table 2 and the reference vectors and distributions in Table 3. We now extend the pseudocode of step 8:

TYPE	Non-Fiction	Fiction
B^{C1}	0	1
B^{C2}	2	1
B^{C3}	1	0
B^{C4}	1	3

ISBN	231	475	523	673	856	937
B^{C1}	0	0	1	0	0	0
B^{C2}	1	0	1	0	1	0
B^{C3}	1	0	0	0	0	0
B^{C4}	0	1	0	1	1	1

Table 2. Vector representation of the bags of the TYPE and ISBN attributes for each target case (C1 to C4) after the exploration in Table 1. As before, the B's denote the counts of how often a value appeared.

TYPE	Non-Fiction	Fiction
B^1	1.5	0.5
B^0	0.5	2.0
B^*	2.0	2.5
D^1	0.75	0.25
D^0	0.20	0.80
D^*	0.44	0.55

ISBN	231	475	523	673	856	937
B^1	1	0	0.5	0	0.5	0
B^0	0	0.5	0.5	0.5	0.5	0.5
B^*	0.5	0.25	0.5	0.25	0.5	0.25
D^1	0.5	0	0.25	0	0.25	0
D^0	0	0.2	0.2	0.2	0.2	0.2
D^*	0.22	0.11	0.22	0.11	0.22	0.11

Table 3. Estimates of the reference distributions for the TYPE and ISBN attributes for the bag of related objects in Table 1: class-conditional positive D^1, class-conditional negative D^0, and unconditional distribution D^*. The corresponding reference bags (B^1, B^0, and B^*) capture the same information, but with a different normalization: division by the number of target cases rather than by the number of related entities.

```
Input: All bags $R_{T_{ga}}(t_t^f)$ for attribute $a \neq j$ of all target cases $t_t^f$.
8.1   Foreach target case $t_t^f$ estimate $B_{T_{ga}}^f$
8.2   Estimate $B_{T_{ga}}^0, B_{T_{ga}}^1, B_{T_{ga}}^*, D_{T_{ga}}^0, D_{T_{ga}}^1, D_{T_{ga}}^*$
```

The aggregation in step 11 now can take advantage of the reference vectors by applying different vector distances between a case vector and a reference vector. An aggregation was defined as a mapping from a bag of values to a single value. We now define *vector-distance aggregates* of a bag of categorical attributes T_{ji} as:

$$A(R_{T_{ji}}(t_t^f)) = VD(B_{T_{ji}}^f, RV) \tag{11}$$

where VD can be any vector distance metric and $RV \in \{B_{T_{ji}}^0, B_{T_{ji}}^1, B_{T_{ji}}^* D_{T_{ji}}^0, D_{T_{ji}}^1, D_{T_{ji}}^*\}$. ACORA offers a number of distances measures for these aggregations: likelihood, Euclidean, cosine, edit, and Mahalanobis, since capturing different notions of distance is one of the principles from Section 2.1. In the case of cosine distance the normalization (B^0 vs. D^0) is irrelevant, since cosine normalizes by the vector length.

Consider (Table 4) the result of step 12 of the algorithm on our example, where two new attributes are appended to the original feature vector in the target table, using cosine distance to B_1 for the bags of the TYPE and the ISBN attributes. Both features appear highly predictive, but of course the predictive

CID	**CLASS**	$Cosine(B^{CID}_{TYPE}, B^{1}_{TYPE})$	$Cosine(B^{CID}_{ISBN}, B^{1}_{ISBN})$
C1	0	0.316	0.408
C2	1	0.989	0.942
C3	1	0.948	0.816
C4	0	0.601	0.204

Table 4. Feature table F after appending the two new cosine distance features from bags of the TYPE and ISBN variable to the class-conditional positive reference bag. The new features show a strong correlation with the class label.

power has to be evaluated in terms of the out-of-sample performance for test cases that were not used to construct B^0 and B^1.

Observe the properties of these operators in light of the principles derived in Section 2.1: 1) they are task-specific if RV is one of the class-conditional reference vectors, 2) they compress the information from categorical attributes of high dimensionality into single numeric values, and 3) they can capture different notions of similarity if multiple vector distance measures are used. If the class labels change, the features also will, since the estimates of the distributions will differ. If there were indeed two different class-conditional distributions, the case vectors of positive examples would be expected to have smaller distances to the positive than to the negative class-conditional distribution. The new feature (distance to the positive class-conditional distribution) will thereby reflect a strong similarity with respect to the task. This can be observed in Table 4. Only if the two class distributions are indeed identical should the difference in the distances be close to zero. The loss of discriminative information is lower compared to conventional aggregation.

A limitation of the features constructed by ACORA is they are not easily comprehensible. The only conclusion that could be drawn about the use by a model of such a vector distance feature is that the distribution of a particular attribute is different for target cases of one class versus the other. In order to understand more fully, it would be necessary to analyze or visualize the actual differences between D^0 and D^1.

The computational complexity of the aggregation considering only one join is $O(n * p * \log p)$, where n is the length of the table after the join and p is the number of possible categorical values. The class-conditional distribution can already be estimated during the join execution. One additional pass over the resulting table is required to estimate the distances. The $p * \log p$ factor reflects the use of hash tables to store intermediate results. n can be approximated as a product of the size of the target table and the average size s of the bag of related objects: $n \sim s_t * s$. The quality of this estimate depends on the variance and the skew of the distribution of the bag sizes. The overall complexity of ACORA is $O(J * n * p * \log p)$ if J joins are considered.

Categorical Counts An alternative solution to address the cardinality and the resulting length of the vector representation B is to curtail the vector by selecting a smaller subset of values for which the counts are recorded. This poses the question of a suitable criterion for selection. A simple selection criterion is the overall frequency of a value across all bags. ACORA constructions in addition to the vector-distance features, the top n values v for which $B^*(O(v))$ was largest.

However, the principles in Section 2.1 suggest choosing the *most discriminative* values for the target prediction task. Specifically, ACORA uses the class-conditional reference vectors B^0 and B^1 (or the distributions D^0 and D^1) to select those that show the largest absolute values for $B^1 - B^0$. For example, the most discriminative TYPE value in the example is 'Fiction with a difference of 1.5 in Table 3.

Numeric Aggregates ACORA provides straightforward aggregates for numeric attributes: MIN, MAX, SUM, $MEAN$, and $VARIANCE$. It also discretizes numeric attributes (equal-frequency binning) and estimates class-condi–tional distributions and distances, similar to the procedure for categorical attributes described in Section 3.1. This aggregation makes no prior assumptions about the distributions (e.g., normality) and can capture arbitrary numeric densities. We do not assess this capability in this paper.

3.2 Computation of Bags of Related Objects

As introduced briefly in Section 3, one component of relational learning is the identification of entities that are related to the observations in the target table. This requires knowledge about the available background tables, the types of their attributes, and which attributes can be used to join. ACORA first distinguishes a set of identifiers using the proposed heuristic that requires an identifier to be categorical and to have cardinality larger than some constant, typically set to 100. Using this set of identifier attributes, ACORA converts a domain explicitly into a graph representation and finds related information using breadth-first search for graph traversal. As an example to illustrate this process we use the CORA domain [9], a bibliographic database of machine learning papers (see Section 7). CORA comprises three tables: Paper, Author and Citation, as shown in Figure 4. We do not use the text of the papers, only the citation and authorship information.

The first step is the conversion of the domain into a graph. The tables are the vertices and an edge between two tables T_j and T_m represents the occurrence of a pair of identifier attributes T_{ji} and T_{ml} that are compatible, i.e., they belong to the equality relation $ER(T_{ji}, T_{ml})$. The only condition imposed on an edge is that T_j and T_m cannot both be equal to the target table T_t. This allows for multiple edges between two tables. With the exception of the target table, we also allow

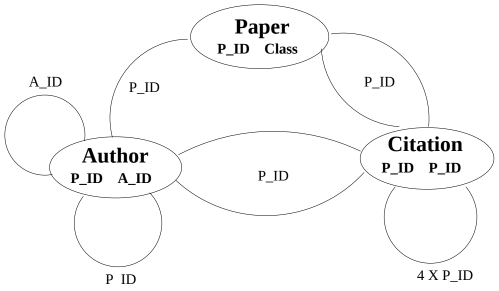

Fig. 4. Graph representation of the CORA document classification domain, with target table Paper(P_ID and Class), two background tables Author(P_ID,A_ID) and Citation(P_ID,P_ID). Each identifier also produces a self-loop, except on the target table.

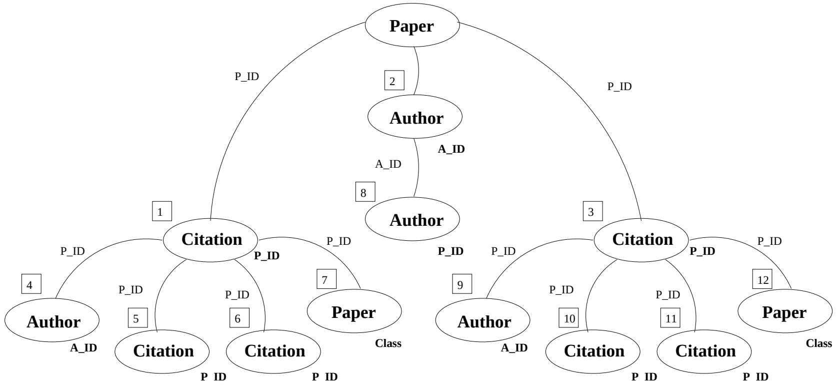

Fig. 5. Dynamic exploration tree corresponding to a breadth-first search over the CORA graph in Figure 4. The exploration starts from the target table Paper. The numbers denote the order in which the nodes are visited; attribute names on links show the identifier that was used for the join, and the attribute names to the right of each node denote attributes that have to be aggergated.

edges that link a table to itself.[4] Figure 4 shows the CORA graph including the target table, Paper, and two additional tables, Author and Citation, showing attributes in the nodes and the linking identifiers on the edges. P_ID and A_ID stand for PaperId and AuthorId respectively, and are identifiers; attributes with the same name have the same type.

ACORA's search explores this domain graph starting from the target table using breadth-first search as formalized in the pseudocode in Figure 3. Figure 5 shows the "unrolled" search tree, the numbers corresponding to the order of breadth-first search. The path from the root to each node of the tree corresponds to a sequence of joins, and so the nodes in layer of depth n represent all possible

[4] Self-links currently are not included for target tables because they cannot provide any new information for the propositional learning task.

joins over n relations. The results of the sequence of joins are the bags of related entities from the final nodes for each object in the target table.

The only constraint on a path is that for any table, the incoming identifier attribute (a particular column, not the type) must be different from the outgoing identifier attribute. The intuition for this heuristic can be seen in the CORA domain: joining the Paper table on P_ID with Author produces for each paper the set of its authors. A second join on P_ID to the citation table would produce for each paper-author pair a bag of all cited papers. Now each citation appears n times where n is the number of authors of the paper. We have only duplicated the information about the citations by a factor of n. Intermediate tables on a path that reuses the same key only result in a replication of information that would be available on a shorter path that skips that table.

Given cycles in the graph, it is necessary to impose a stopping criterion. ACORA uses either depth (three in the case of Figure 5) or the number of joins. As the length of a path increases, the distance to the target object increases and the relevance of those related entities decreases. Alternative stopping criteria include the number of constructed features, run time, minimum gain in model performance, etc.

Finally we have to decide whether ACORA should be permitted to join back to the target table (see nodes 6, 9 and 14 in Figure 5) and if yes, under what conditions. This question is related to the definition of what constitutes background knowledge for a particular problem. More specifically, is the collection of training data itself part of the background knowledge that will be available for prediction? This view is often appropriate for networked domains ([10],[11],[12]).

4 Formal Analysis and Implications

We suggested distance-based aggregates to address a particular problem: the aggregation of categorical variables of high cardinality. The empirical results in Section 5 provide support that distribution-based aggregates can indeed condense information from such attributes and improve generalization performance significantly over alternative aggregates, such as counts for the n most common values. We now show that the distance-based aggregation operators can be derived as components of a "relational fixed-effect model" with a Bayesian theoretical foundation. This derivation also allows us to identify assumptions that impact the performance and to derive (see Section 4.3) implications for the use of aggregation with identifiers.

4.1 Distributional Meta-Data for Aggregation

Aggregation summarizes a set or a distribution of values. As we have described, ACORA creates reference summaries, and saves them as "meta-data" about the unconditional or class-conditional distributions, against which to compare summaries of the values related to particular cases. Specifically, the normalized reference vectors D_1 and D_0 are the class-conditional likelihoods that define the distribution from which the values in the corresponding bag would have been sampled, under the assumption of independent draws.

Although its use is not as widespread as in statistical hypothesis testing, distributional meta-data like D_1 and D_0 are not foreign to machine learning. Naive Bayes stores class-conditional likelihoods for each attribute. In fraud detection, distributions of normal activity have been stored, to produce variables indicating deviations from the norm [13]. Aggregates like the mean and the standard deviation of related numeric values also summarize the underlying distribution; under the assumption of normality those two aggregates fully describe the distribution. Even the $MODE$ of a categorical variable is a crude summary of the underlying distribution (i.e., the expected value). In the case of categorical attributes, the distribution can be described by the likelihoods—the counts for each value normalized by the bag size. So all these aggregators attempt to characterize for each bag the distribution from which its values were drawn. Ultimately the classification model using such features tries to find differences in the distributions.

In principle, each object has an associated distribution from which the values are drawn. The methodology of estimating the likelihoods for categorical attributes is clear; however, estimating these distributions from a bag of categorical values of a high-cardinality attribute is problematic. The number of parameters (likelihoods) for each distribution is equal to the attribute's cardinality minus one. Unless the bag of related entities is significantly larger than the cardinality, the estimated likelihoods will not be reliable: the number of parameters often will exceed the size of the bag.[5] We make the simplifying assumption that all objects related to any positive target case were drawn from the **same** distribution. We therefore only estimate two distributions, rather than one for each target case. A similar distinction has been made in traditional statistical estimation.

4.2 A Relational Fixed-Effect Model

Statistical estimation contrasts *random-effect* models from *fixed-effect* models [14]. In a random-effect model, model parameters are not assumed to be constant but instead to be drawn from different distributions for different observations. Estimating one distribution for each bag corresponds to a random effect model. Our aggregates on the other hand implement a relational *fixed-effect* model. We assume one fixed distribution for each of the two classes. Under this assumption the number of parameters decreases by a factor of $n/2$ where n is the number of training examples. More specifically, the main assumption for a *relational* fixed-effect model is that all bags of objects related to positive target cases are sampled from one distribution D_1 and all objects related to negative target cases are drawn from another distribution D_0. Thus it may become possible to

[5] The same problem of too few observations can arise for numeric attributes, if the normality assumption is rejected and one tries to estimate arbitrary distributions (e.g., through Gaussian mixture models).

compute reliable estimates of reference distributions D_1 and D_0 even in the case of categorical attributes of high cardinality, by combining all bags related to positive/negative cases to estimate D_1/D_0.

Even with only two distributions it still is necessary to construct features for the bag(s) of values related to each case. ACORA computes these with various vector distances. Notably, the *likelihood* of observing a particular bag of values assuming a class-conditional distribution from which they were independently sampled can be seen as a particular vector distance (where i ranges over the set of possible values for the bagged attribute):

$$LH(B, D_c) = \frac{1}{\prod_i D_0[i]^{B[i]} + \prod_i D_1[i]^{B[i]}} \prod_i D_c[i]^{B[i]} \tag{12}$$

For the particular choice of likelihood as the distance function, the relational fixed-effect model can be given a theoretical foundation within a general relational Bayesian framework very similar to that of Flach and Lachiche ([15],[16]). In a relational context, a target object t_t is not only described by its attributes, but it also has an identifier (CID in our example) that maps into bags of related objects from different background tables. Starting with Bayes' rule one can express the probability of class c for a target object t_t with a feature vector[6] and a set of bags of related objects from different relationships $(t_{t1}, \ldots, t_{tk}, R_{T_u}(t_t), \ldots, R_{T_v}(t_t))$ as

$$P(c|t_t) = P(c|t_{t1}, \ldots, t_{tk}, R_{T_u}(t_t), \ldots, R_{T_v}(t_t)) \tag{13}$$

$$= P(t_{t1}, \ldots, t_{tk}, R_{T_u}(t_t), \ldots, R_{T_v}(t_t)|c) * P(c)/P(t_t). \tag{14}$$

Making the assumption of class-conditional independence of the attributes and of the bags of related objects allows rewriting the above expression as

$$P(c|t_t) = \prod_i P(t_{ti}|c) * \prod_j P(R_{T_j}(t_t)|c) * P(c)/P(t_t). \tag{15}$$

Assuming that the elements t_j in the a bag $R_{T_j}(t_t)$ are drawn independently, we can rewrite

$$P(R_{T_j}(t_t)|c) = \prod_{t_j^f \in R_{T_j}(t_t)} P(t_j^f|c). \tag{16}$$

Assuming again class-conditional independence of the attributes t_{j*} of related entities, we can finally estimate the class-conditional probability of a bag object from the training data as

$$P(R_{T_j}(t_t)|c) = \prod_{t_j^f \in R_{T_j}(t_t)} \prod_m P(t_{jm}^f|c). \tag{17}$$

Switching the order of the product this term can be rewritten as a product over all attributes over all samples:

$$P(R_{T_j}(t_t)|c) = \prod_m \prod_{t_{jm}^f \in R_{T_{jm}}(t_t)} P(t_{jm}^f|c). \tag{18}$$

[6] Excluding the class label and the identifier.

This non-normalized (not accounting for $P(c)$ and $P(t_t)$) probability $P(R_{T_jm}(t_t)|c)$ corresponds directly to our distance-based aggregate that uses the likelihood distance from equation 12 between the case vector B and D_c since $D_c[O(v)]$ of value v is an unbiased estimate of the class conditional probability $P(v|c)$:

$$D_c[O(t^f_{jm})] \hat{=} P(t^f_{jm}|c) \quad \Rightarrow \quad \prod_i D_c[i]^{B[i]} \hat{=} \prod_{t^f_{jm} \in R_{T_{jm}}(t_t)} P(t^f_{jm}|c) \tag{19}$$

This derivation provides one theoretical justification for our more general framework of using (multiple) vector distances in combination with class-conditional distribution estimates. It also highlights the three inherent assumptions of the approach: 1) class conditional independence between attributes (and identifiers) of the target cases, 2) class-conditional independence between related entities, and 3) class conditional independence between the attributes of related objects. Strong violations are likely to decrease the predictive performance. It is straightforward to extend the expressiveness of ACORA to weaken the first assumption, by (for example) combining pairs of feature values prior to aggregation. The second assumption, of random draws, is more fundamental to aggregation in general and less easily addressed. Relaxing this assumption comes typically at a price: modeling becomes increasingly prone to overfitting because the search space expands rapidly. This calls for strong constraints on the search space, as are typically provided for ILP systems in the declarative language bias. We discussed this tradeoff previously [2] in the context of noisy domains.

4.3 Learning from Identifier Attributes

We show in our empirical results in Section 5 the importance of including aggregates of identifiers. The following discussion is a somewhat formal analysis of the special properties of identifiers and why aggregates of identifiers and in particular additive distances like cosine can achieve such performance improvements.

We defined identifiers as categorical attributes with a high cardinality. In our example problem we have two such attributes: CID, the identifier of customers, and ISBN, the identifier of books. Considering the task of classifying customers based on the target table T_t clearly calls for the removal of the unique CID attribute prior to model induction, because it cannot generalize. However, the identifiers of *related* objects may be highly predictive out-of-sample (e.g., anybody who has met with Bin Laden is very likely to be in involved in terrorist activity), because they are shared across multiple target cases that are related to the same objects (e.g., customers who bought the same book). The corresponding increase in the effective number of appearances of the related-object identifier attribute T_{kj}, such as ISBN, allows the estimation of class-conditional probabilities $P(t_{kj}|c)$.

Beyond the immediate relevance of particular identities (e.g., Bin Laden), identifier attributes have a special property: they represent implicitly all characteristics of an object. Indeed, the *identity* of a related object (such as Bin Laden) can be more important than any set of available attributes describing that object. This has important implications for modeling: using identifier attributes can overcome the limitations of class-conditional independence in Equation 16 and even permits learning from unobserved characteristics.

An object identifier t_{kj} like ISBN stands for all characteristics of the object. If observed, these characteristics would appear in another table T_m as attributes $(t_{m1},..,t_{mn})$. Technically, there exists a functional mapping[7] F that maps the identifier to a set of values: $F(t_{kj}) \rightarrow (t_{m1},..,t_{mn})$. We can express the joint class-conditional probability (without the independence assumption) of a particular object feature-vector without the identifier attribute as the sum of the class-conditional probabilities of all objects f with the same feature vector:

$$P(t_{m1},..,t_{mn}|c) = \sum_{f:F(t^f_{kj})=(t_{m1},..,t_{mn})} P(t^f_{kj}|c) \tag{20}$$

If F is an isomorphism (i.e., no two objects have the same feature vector) the sum disappears and $P(t_{m1},..,t_{mn}|c) = P(t_{kj}|c)$. Estimating $P(t_{kj}|c)$ therefore provides information about the joint probability of all its attributes $(t_{m1},..,t_{mn})$.

A similar argument can be made for an unobserved attribute t_{mu} (e.g., membership in a terrorist organization). In particular it may be the case that no attribute of the object t_{kj} was observed and no table T_m was recorded, as is the case for ISBN in our example. There is nevertheless the dependency $F'(t_{kj}) \rightarrow t_{mu}$, for some function F', and the relevant class-conditional probability is equal to the sum over all identifiers with the same (unobserved) value:

$$P(t_{mu}|c) = \sum_{f:F'(t^f_{kj})=t_{mu}} P(t^f_{kj}|c). \tag{21}$$

Given that t_{mu} is not observable, it is impossible to decide which elements belong into the sum. If however t_{mu} is a perfect predictor—i.e. every value of t_{mu} appears only for objects related to target cases of one class c—the class-conditional probability $P(t^f_{kj}|c)$ will be non-zero for only one class c. In that case the constricted sum in Equation 21 is equal to the total sum over the class-conditional probabilities of all identifier values:

$$\sum_{f:F'(t^f_{kj})=t_{mu}} P(t^f_{kj}|c) = \sum_f P(t^f_{kj}|c). \tag{22}$$

Note that the total sum over the class-conditional probabilities of all related identifier values now equals the cosine distance between D_c and a special case vector B^{all} that correspond to a bag containing all identifiers with value t_{mu} prior to normalization[8] is by vector length, since $D^c[O(t^f_{kj})]$ is an estmiate of

[7] This function F does not need to be known; it is sufficient that it exists.

[8] The effect of normalization can be neglected, since the length of D_c is 1 and the length of B is the same for both the class-conditional positive and class-conditional negative cosine distances.

$P(t^f_{kj}|c)$ and $B[O(t^f_{kj})]$ is typically 1 or 0 for identifier attributes such as ISBN. The cosine distance for a particular bag B^{t_t} is a biased estimate of $P(t_{mu}|c)$ since the bag will typically only consist of a subset of all identifiers with value t_{mu}[9].

$$cosine(D^c_{t_{kj}}, B) = \frac{1}{||B||} \sum_i D^c_{t_{kj}}[i] * B_{t_{kj}}[i] \tag{23}$$

So far we have assumed a perfect predictor attribute T_{mu}. The overlap between the two class conditional distributions D^0 and D^1 of the identifier is a measure of the predictive power of T_{mu} and also how strongly the total sum in the cosine distance deviates from the correct restricted sum in Equation 21. The relationship between the class-conditional probability of an unobserved attribute and the cosine distance on the identifier may be the reason why the cosine distance performs better than likelihood in the experiments in Section 5.

Although this view is promising, issues remain. It often remains hard to estimate $P(t_{kj}|c)$ due to the lack of sufficient data (it is also much harder to estimate the joint rather than a set of independent distributions). We often do not want to estimate the entire joint distribution because the true concept is an unknown class-conditional dependence between only a few attributes. Finally the degree of overlap between the two class-conditional distributions D^0 and D^1 determines how effectively we can learn from unobserved attributes.

Nevertheless, the ability to account for identifiers through aggregation can extend the expressive power significantly as shown empirically in Section 5. Identifiers have other interesting properties. They may often be the *cause* of relational auto-correlation ([17]). Because a customer bought the first part of the trilogy, he now wants to read how the story continues. Given such a concept, we expect to see auto-correlation between customers that are linked through books.

In addition to the identifier proxying for all object characteristics of immediately related entities (e.g., the authors of a book), it also contains the implicit information about all other objects linked to it (e.g., all the other books written by the same author). An identifier therefore introduces a “natural” Markov barrier that reduces or eliminates the need to extend the search for related entities further than to the direct neighbors. We present some evidence of this phenomenon in Section 5.3.

5 Empirical Results

We introduced distribution-based aggregates in order to construct features for relational domains where exploration of the relational structure will yield bags of values from categorical attributes of high cardinality. After introducing the experimental setup, Section 5.3 presents the empirical evidence in support of our main claims regarding the generalization performance of the new aggregates. Then we present a sensitivity analysis of the factors influencing the results (Section 5.4).

[9] We underestimate $P(t_{mu}|c)$ as a function of the size of the bag. The smaller the bag, the more elements of the sum are 0 and the larger the bias.

Domain	Table: Size	Attribute Type Description	Size
XOR	T: 10000	C(tid)=10000, C(c)=2	Train:8000
	O: 55000	C(oid)=10000, C(tid)=10000	Test: 2000
AND	T: 10000	C(tid)=10000, C(c)=2	Train:8000
	O: 55000	C(oid)=10000, C(tid)=10000	Test: 2000
Fraud	T: 100000	C(tid)=100000	Train:50000
	R: 1551000	C(tid)=100000, C(tid)=100000	Test: 50000
KDD	T: 59600	C(tid)=59600, C(c)=2	Train:8000
	TR: 146800	C(oid)=490, C(tid)=59600	Test: 2000
IPO	T: 2790	C(tid)=2790, C(e)=6, C(sic)=415, C(c)=2	Train:2000
		D(d,s,p,r)=ℝ	Test: 800
	H: 3650	C(tid)=2790, C(bid)=490	
	U: 2700	C(tid)=2790, C(bid)=490	
COOC	T: 1860	C(tid)=1860 C(c)=2	Train:1000
	R: 50600	C(tid)=1860 C(tid)=1860	Test: 800
CORA	T: 4200	C(tid)=4200, C(c)=2	Train:3000
	A: 9300	C(tid)=4200, C(aid)=4000	Test: 1000
	R: 91000	C(tid)=4200, C(tid)=35000	
EBook	T: 19000	C(tid)=19000, C(c,b,m,k)=2, D(a,y,e)=ℝ	Train:8000
	TR: 54500	C(oid)=22800, C(tid)=19000, D(p)=ℝ, C(c)=5	Test: 2000

Table 5. Summary of the properties of the eight domains, including the tables, their sizes, their attributes, types, and the training and test sizes used in the main experiments. C(y) is the cardinality of a categorical attributes and D(y)=ℝ identifies numeric attributes.

5.1 Domains

Our experiments are based on eight relational domains that are described in more detail in the Appendix. They are typical transaction or networked-entity domains with predominantly categorical attributes of high cardinality. The first two domains (XOR and AND) are artificial, and were designed to illustrate simple cases where the concepts are based on (combinations of) unobserved attributes. Variations of these domains are also used for the sensitivity analysis later. Fraud is also a synthetic domain, designed to represent a real-world problem (telecommunications fraud detection), where target-object identifiers (particular telephone numbers) have been used in practice for classification [13][18]. The remaining domains include data from real-world domains that satisfy the criteria of having interconnected entities. An overview of the number of tables, the number of objects, and the attribute types is given in Table 5. The equality relation of the types is implied by identical attribute names.

5.2 Methodology

Our main objective is to demonstrate that distribution-based vector distances for aggregation generalize when simple aggregates like *MODE* or *COUNTS* for all values are inapplicable or inadequate. In order to provide a solid baseline we extend these simple aggregates slightly for use in the presence of attributes with high cardinality: ACORA constructs $COUNTS$ for the 10 most common values (an extended $MODE$) and counts for all values if the number of distinct values is at most 50 as suggested by Krogel and Wrobel [19]. ACORA generally includes an attribute representing the bag size as well as all original attributes from the target table.

Feature construction: Table 6 summarizes the different aggregation methods. ACORA uses 50% of the training set for the estimation of class-conditional reference vectors and the other 50% for model estimation. The model estimation cannot be done on the same data set that was used for construction, since the use of the target during construction would lead to overestimation of the predictive performance. We also include distances from bags to the unconditional distribution (estimates calculated on the full training set). Unless otherwise noted, for the experiments the stopping criterion for the exploration is depth = 1, meaning for these domains that each background table is used once. The cutoff for identifier attributes I_{MIN} was set to 400.

Model estimation: We use WEKA's logistic regression [8] to estimate probabilities of class membership from all features. Using decision trees (including the differences of distances as suggested in Section 5.3) did not change the relative performance between different aggregation methods significantly, but generally performed worse than logistic regression. We did not use feature selection for the presented results; feature selection did not change the relative performance, since for these domains the number of constructed features remains relatively small.

Evaluation: The generalization performance is evaluated in terms of the area under the ROC curve (AUC) [20]. All results represent out-of-sample generalization performance on test sets averaged over 10 runs. The objects in the target table are for each run split randomly into a training set and a test set (cf., Table 5). We show error bars of ± one standard deviation in the figures and include the standard deviation in the tables in parentheses.

5.3 Main Results

We now analyze the relative generalization performance of different aggregation operators. Our main claim that class-conditional, distribution-based aggregates add generalization power to classification modeling with high-dimensional categorical variables was motivated by four arguments that are considered in the sequel:

- Target-dependent aggregates such as vector distances to class-conditional reference vectors exhibit task-specific similarity;
- The task-specific similarity improves generalization performance;
- Aggregating based on vector distances condenses discriminative information from identifier attributes;
- The use of multiple aggregate operators for the same bag improves generalization performance.

Method	**Description**
COUNTS	ACORA constructs count features for all possible categorical values if the number of values is less than 50. In particular this excludes all key attributes.
MCC	Counts for the 10 most common categorical values (values with largest entries in unconditional reference bag B^*). MCC can be applied to all categorical attributes including identifiers.
MDC	Counts for the 10 most discriminative categorical values (Section 3.1) defined as the values with the largest absolute difference in the vector $B^1 - B^0$. MDC can be applied to all categorical attributes including identifiers.
Cosine	Cosine(D^1, B^{t_t}), Cosine(D^0, B^{t_t})
Mahalanobis	Mahalanobis(B^1, B^{t_t}), Mahalanobis(B^0, B^{t_t})
Euclidean	Euclidean(B^1, B^{t_t}), Euclidean(B^0, B^{t_t})
Likelihood	Likelihood(D^1, B^{t_t}), Likelihood(D^0, B^{t_t})
UCVD	All unconditional distances: Cosine(D^*, B^{t_t}), Mahalanobis(D^*, B^{t_t}), Euclidean(D^*, B^{t_t}), Likelihood(D^*, B^{t_t})
CCVD	All class-conditional distances: Cosine(D^1, B^{t_t}), Cosine(D^0, B^{t_t}), Euclidean(D^1, B^{t_t}), Euclidean(D^0, B^{t_t}), Mahalanobis(D^1, B^{t_t}), Mahalanobis(D^0, B^{t_t}), Likelihood(D^1, B^{t_t}), Likelihood(D^0, B^{t_t})
DCCVD	All differences of class-conditional distances: Cosine(D^1, B^{t_t}) – Cosine(D^0, B^{t_t}), Mahalanobis(D^1, B^{t_t}) – Mahalanobis(D^0, B^{t_t}), Euclidean(D^1, B^{t_t}) – Euclidean(D^0, B^{t_t}), Likelihood(D^1, B^{t_t})– Likelihood(D^0, B^{t_t})

Table 6. Summary of aggregation operators used in the experiments, grouped by type: counts for particular categorical values, different vector distances, combinations of vector distances to conditional or unconditional reference distributions where t_t denotes a target case.

Task-Specific Similarity We argued in Section 2.1 that task-specific aggregates have the potential to identify discriminative information because they exhibit task-specific similarity (making positive instances of related bags similar to each other). Figure 6 shows for the XOR problem the two-dimensional instance space defined by using as attributes two class-conditional aggregations of identifiers of related entities: the cosine distance to the positive distribution and the cosine distance to the negative distribution. Although the positive target objects each had a different bag of identifiers, using the constructed attributes the positive objects are similar to each other (left-upper half) and the negative are similar to each other (right-lower half).

Importantly, it also is clear from the figure that although positive target cases have on average a larger cosine distance to the positive class-conditional distribution (they are mostly on the left side of the plot) than negative cases, only the combination of both features becomes very discriminative between the two

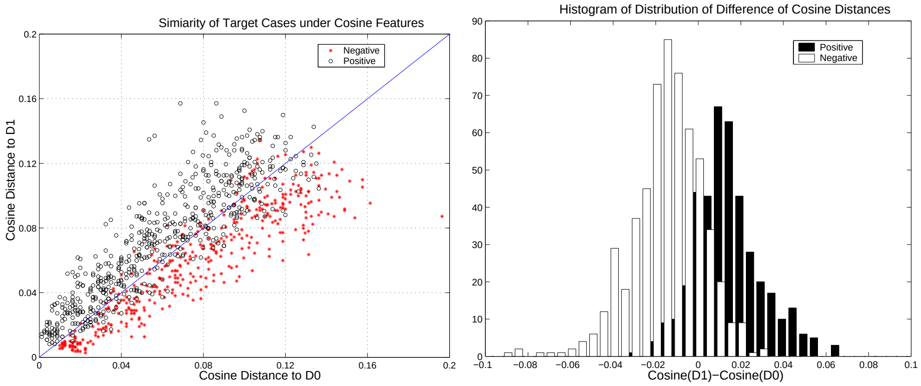

Fig. 6. In the left plot, the two-dimensional feature space (XOR domain) of the class-conditional cosine distances for the identifiers of related entities shows high instance-discriminability (different target cases are assigned unique points in this space) and task-specific similarity, where negative cases are grouped on the lower right of the identity line and positive target cases on the upper right. This similarity leads to a high class-discriminability using the identity line as decision boundary. In the right plot, after a transformation of the feature space that takes the difference between class-conditional cosine distances, the distribution of the new feature shows a good class separation. This transformation is of particular value for model induction using decision trees, which make axis-parallel splits, and for feature selection in order to ensure that the joint predictive information of both distances is preserved.

classes. In fact, there is an approximate linear decision boundary (the diagonal), which implies that logistic regression would be a good choice for model induction. For a decision tree, with axis-parallel splits, the difference between the two distances is a better feature. Figure 6 shows on the right the distribution of the differences for cases of both classes with an optimal splitting point around zero. This explains the better performance we will see later of a decision tree using DCCVD in Figure 9 over the individual positive and negative distances CCVD.

Figure 7 on the other hand shows the feature space of unconditional cosine and Euclidean distances. These task-independent features do not provide discriminative information. Positive and negative cases are mixed, and in particular are not more similar to each other than to cases of the opposite class.

Comparative Generalization Performance We now show that the use of aggregations based on distributional meta-data adds generalization power over traditional aggregations (and our extensions to the traditional methods). Table 7 presents the generalization performance (AUC) of the different aggregation strategies across all domains. First, consider the second and third columns. These correspond to the (extended) traditional aggregations: value-count features (COUNTS) and most-common-value features (MCC). Because of the high dimensionality of the categorical attributes, COUNTS features simply are inap-

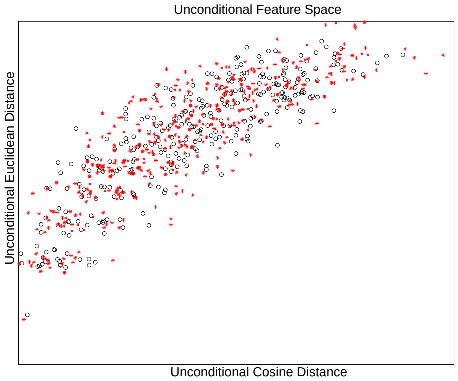

Fig. 7. The two-dimensional feature space of the unconditional cosine and Euclidean distances still shows high instance-discriminability, but lacks task-specific similarity. Positive cases are as similar to negative cases as they are to other positive cases. As a result these features have no discriminative power.

plicable in most of the domains. (Entries with a * denote cases where a COUNTS aggregation was not applicable because all categorical attributes had too many distinct values and no features were constructed.) For IPO, the AUC nevertheless is greater than 0.5 because in this domain the target table had attributes for propositional modeling. Ebook is the only domain where COUNTS aggregates are applicable and add generalizability.

The fourth through sixth columns correspond to the construction of different sorts of distribution-based aggregations (respectively, unconditional, class-conditional, and most-discriminative counts). For all domains the aggregation of high-dimensional categorical attributes using class-conditional distributions (CCVD) leads to models with relatively high generalization performance (AUC scores between 0.78 and 0.97). In all but one case (the tie with MDC on IPO) the features based on class-conditional distributions perform better—often significantly better—than those based on unconditional distributions and those based on most-discriminative counts. Finally, combining MDC and CCVD (reported in the seventh column) improved the performance over CCVD only slightly on three domains (COOC, EBook and IPO).

Recall the two main components of the design of the CCVD aggregations: their task-specific (class-conditional) nature and their incorporation of information from many values (using distribution distances). The consistently superior performance of class-conditional distribution distances over unconditional distribution distances highlights the importance of task-specific aggregation, which also is seen clearly in the often-improved performance of counts of most-discriminative values (MDC) over counts of most common values (MCC). The consistently superior performance of CCVD over MDC highlights the importance of considering the entire distributions, more fully satisfying the design principles.

Domain	COUNTS	MCC	UCVD	CCVD	MDC	MDC&CCVD
XOR	0.5*	0.51 (0.004)	0.62 (0.02)	0.92 (0.008)	0.51 (0.004)	0.92 (0.008)
AND	0.5*	0.52 (0.012)	0.65 (0.02)	0.92 (0.006)	0.52 (0.007)	0.92 (0.05)
Kohavi	0.5*	0.71 (0.022)	0.72 (0.024)	0.85 (0.025)	0.84 (0.044)	0.85 (0.025)
IPO	0.70* (0.023)	0.77 (0.02)	0.75 (0.021)	0.79 (0.03)	0.79 (0.003)	0.82 (0.01)
CORA	0.5*	0.74 (0.018)	0.67 (0.008)	0.97 (0.003)	0.76 (0.008)	0.97 (0.006)
COOC	0.5*	0.63 (0.016)	0.57 (0.017)	0.78 (0.02)	0.63 (0.02)	0.80 (0.04)
EBook	0.716 (0.024)	0.79 (0.011)	0.88 (0.015)	0.95 (0.024)	0.94 (0.018)	0.96 (0.013)
Fraud	0.5*	0.49 (0.005)	0.74 (0.020)	0.87 (0.028)	0.51 (0.006)	0.87 (0.021)

Table 7. Comparison of generalization performance (AUC) for different aggregation strategies (see Table 6 for a description). Entries with * denote cases where the COUNTS aggregation was not applicable because all categorical attributes had too many distinct values. The standard deviation across 10 experiments is included in parenthesis.

For the artificial domains and the synthetic fraud domain, neither type of most-common count (MCC nor MDC) provides any predictive power. This will be explained in Section 5.4. For the COOC domain, on the other hand, the most common tickers related to technology firms and the most discriminative tickers related to technology firms happen to be the same: GE, MSFT, CSCO, IBM, AOL, INTC, ORCL, AMD, LU, SUNW.

Learning from Identifier Attributes In our collection of domains, identifiers are the main source of information. The only domain with related entities with additional information besides the identifier is EBook. Table 7 not only shows the superiority of feature construction based on class-conditional distributions, but also that it is commonly possible to build highly predictive relational models from identifiers. To our knowledge, this has not been shown before in any comprehensive study. It is important because identifiers often are used only to identify relationships between entities but not directly as features when building predictive models.

We argue in Section 4.3 that identifiers can allow learning from concepts that violate class-conditional independence and from unobserved properties. Our results provide some support for this claim. In the synthetic domains AND and XOR the true concept was a function of two "unobserved" attributes x, y. Therefore that AUC = 0.92 for CCVD for both AND and XOR strongly supports the claim that aggregating identifiers allows learning from unobserved attributes. Even if the the values are provided, these domains violate the model's assumption of class-conditional independence. Consider in addition to the performances in Table 7 the performance of COUNTS if the two attributes x and y were included: 0.5 for XOR and 0.97 for AND. For XOR the independent information about the bags of x's and y's is not at all informative about the class. For AND on the other hand, observing a large number of 1's for x and also a large number of 1's for y increases the probability that the majority of related entities have both $x = 1$ and $y = 1$ (the true concept). The XOR domain provides an example where the aggregation of identifier attributes mitigates the effect of violations of class-conditional independence.

For further evidence we examine the Fraud domain. The underlying concept is that fraudulent accounts call numbers that were previously called by (now known) fraudulent accounts. A model should perform well if it identifies accounts that have two-hop-away fraudulent neighbors. Therefore, ACORA should construct a feature at search depth two, aggregating the class labels of those entities. However, so far we have restricted the search to depth 1. The results in Table 7 therefore indicate that it is possible to classify fraud already from the direct neighbors—similar to the "dialed-digit" monitor reported as a state-of-the-art fraud detection method [13]. Exploring the two-hop-away neighbors and their class labels increases the ranking performance only minimally—to 0.89 compared to 0.87. This suggests that identifiers proxy not only for the (perhaps latent) properties of the object, but also for the other objects to which it is related. We now investigate this further.

Even if the identifiers capture properties of further related entities, it may still be of advantage to explore beyond depth 1 explicitly. The search may 1) find attributes that drive the underlying concept directly (e.g., the fraud label of two-hop-away accounts) and 2) improve the quality of the estimated class-conditional distributions. For (2), if paths comprise sequences of one-to-many relationships, as in the fraud case, the average bag size (average number of phone calls) increases with every new join. In the fraud domain, the branching factor of 20 implies an average of 400 two-hop-away connections for each target case. The estimation of the 100000 parameters of the distributions will be better from a total of 400*25000 (25000 is half of training size) "effective" data points than from 20*25000. On the other hand, the discriminative power (difference between the two class-conditional distributions) will decrease with the number of joins; eventually all target objects (positive and negative) are related to all entities.

Figures 8 show the ranking performance of class-conditional cosine distances as a function of the number of joins for two different training sizes on the CORA and IPO domains. The quality of the estimates of the distributions should be lower for small training sizes and might therefore profit more from a deeper exploration. Indeed, for the IPO domain with the smaller training size, deeper exploration helps. This suggests that the estimates of the distributions improve with a larger effective number of cases (in particular, since the average number of related entities in the joins of the first level was only 2; see Table 11). However, for all other cases we see the performance flatten out after 2 to 4 joins (depth=1 or 2), supporting the claim that the identifiers capture information not only of the entity itself, but also of related entities and in particular of their class labels.

Use of Multiple Aggregates In Section 2.1 we advocated the use of multiple aggregates to capture different notions of similarity. Figure 6 in Section 5.3 already shows the importance of using cosine distances both to the positive

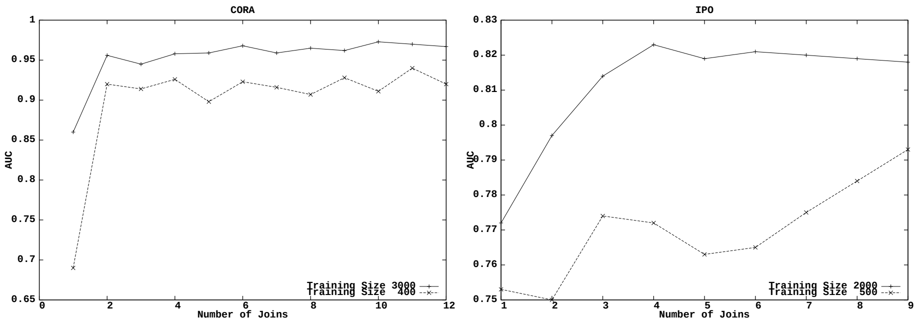

Fig. 8. Ranking performance (AUC) on the CORA (left) and IPO (right) domains as a function of the number of joins for two different training sizes (400 and 3000). Beyond depth=1 (see Figure 5) no new discriminative information is found for CORA, because the depth-one identifier attributes capture information about all objects related further away. For IPO, the maximum performance is reached on the big dataset after 4 joins (corresponding to depth=2). The smaller training size shows performance gains for further joins mostly due to improvements of the quality of the estimates of the class-conditional distributions, because larger search depth increases the bag size and thereby the effective number of observations.

and to the negative distribution. They capture orthogonal information that in combination is more discriminative than the two individual distances.

Table 8 compares the individual performances of difference distance measures as well as their combined performance. Importantly, combining all distances in CCVD improves only marginally over cosine on the IPO, Fraud, and CORA domains, and even hurts slightly for COOC. Cosine performs almost consistently best with the exception of the Fraud domain where Mahalanobis and Euclidean are slightly better. The Euclidean distance is often competitive, with the exceptions of KDD and COOC. The Mahalanobis distance has good results for Fraud and EBook, but is otherwise dominated by cosine and Euclidean.

Likelihood performs acceptably on the IPO domain (it improves over the propositional performance of 0.7 using only the attributes in the root table), but fails on the other domains. This might be caused by the inherently additive nature of Equations 20 and 21, or by the fact that even for the relational fixed-effect model many identifiers appear in only one bag of related (training) entities and the class-conditional estimate is therefore 0 and requires some correction (e.g., Laplace). Large numbers of corrected entries that randomly appeared for one class or the other obscure the true signal. Another semantic property of likelihood is that it considers only *occurrences* of values as evidence, but not the fact that a value did not occur (which are used as evidence for the other distance measures). Likelihood also tends to produce probabilities that are biased strongly towards 0 or 1 as the size of the bag increases, due to the violation of the independence assumption. This does not affect the classification accurracy but may harm the ranking performance as measured by AUC.

Domain	**Cosine**	**Mahalanobis**	**Euclidean**	**Likelihood**	**CCVD**
XOR	0.91 (0.011)	0.87 (0.014)	0.92 (0.012)	0.75 (0.02)	0.92 (0.008)
AND	0.92 (0.018)	0.68 (0.02)	0.91 (0.026)	0.63 (0.02)	0.92 (0.006)
KDD	0.85 (0.026)	0.65 (0.03)	0.77 (0.029)	0.59 (0.05)	0.85 (0.025)
CORA	0.96 (0.004)	0.89 (0.008)	0.91 (0.015)	0.52 (0.03)	0.97 (0.006)
IPO	0.77 (0.01)	0.74 (0.027)	0.77 (0.012)	0.74 (0.025)	0.79 (0.03)
COOC	0.80 (0.013)	0.61 (0.013)	0.70 (0.021)	0.53 (0.025)	0.78 (0.018)
EBook	0.95 (0.026)	0.94 (0.018)	0.92 (0.016)	0.67 (0.032)	0.96 (0.025)
Fraud	0.84 (0.019)	0.87 (0.021)	0.88 (0.018)	0.62 (0.026)	0.87 (0.010)

Table 8. Comparison of generalization performance (AUC) for different vector distance measures (see Table 6 for further description). The standard deviation across 10 expriments is included in parenthesis.

Finally, Figure 9 shows learning curves for a decision tree on the IPO domain for different distribution distance aggregates with a larger exploration depth (2) than the experiments in Table 7. The curves show that taking the difference (DCCVD) between the positive and negative distances performs better than using them separately (CCVD). The reason is the linear decision boundary in the feature space as shown in Section 5.3.

In summary, there is no single best distance measure, but cosine performs well most consistently. Using multiple vector-distance measures at best improves the performance only marginally over the best distance, but is a very consistent top-performer (even more so than cosine). Note that adding distance measures multiplies the number of constructed features, which may hurt generalization performance especially in domains with many attributes. Including counts for most discriminative values (MDC in Table 7) improves over using only vector distances in some domains (IPO, COOC, and EBook), but only minimally. The use of both positive and negative cosine distances (as done for all distances in the Table 8) almost always improves the ranking results (not shown here) as argued previously in Section 5.3.

Reflecting on our design principles, the experimental evidence supports the conclusion that the good ranking performance across the eight domains is due mostly to the *combination* of target-specificity and instance discriminability, while maintaining a low dimensionality. MDC also reduces dimensionality (although not as strongly) and is target-specific, but instance discriminability is lower than for cosine distance. The other principle of using multiple aggregates with different similarities seems to be helpful, but less important.

5.4 Sensitivity Analysis

There are several properties of domains that have the potential to affect the ability of distribution-based aggregations to capture discriminative information. In particular, noise in class labels, the number and connectivity distribution of related objects, and the amount of data available. We now present several brief

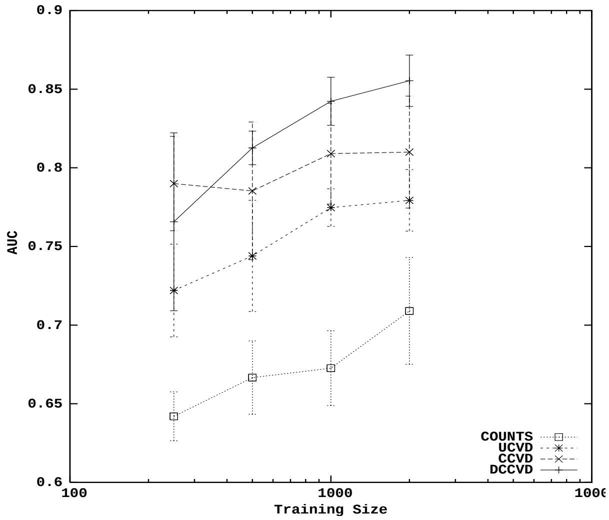

Fig. 9. Ranking ability (AUC) as a function of training size of different reference distributions (counts, unconditional, conditional, conditional difference) on the IPO domain using a decision tree for model induction (standard deviation across 10 experiments is included). The advantage of taking differences DCCVD over CCVD for decision trees (see Section 5.3) is caused by the restriction to axis-parallel splits.

studies illustrating limitations on the applicability of the methods (as well as areas of superior performance).

Noise By class noise we mean that the target classes are not known with perfect accuracy. Class noise will disrupt the accurate estimation of the class-conditional distributions, and therefore may be suspected to lead to degraded performance. For example, consider the use of identifiers to stand in for unobserved attributes (as argued above). In the presence of class noise, using the identifiers may perform considerably worse than if the attributes had been known—because the dimensionality of the unobserved attributes is much smaller and therefore there are fewer parameters to estimate from the noisy data.

We can illustrate this with the AND domain. Recall that from the discussion of the identifier attributes above, aggregation based on COUNTS considering x and y values of related entities performed very well ($AUC = 0.97$). Aggregation using only the identifiers of related attributes (using CCVD) did not perform quite as well ($AUC = 0.92$), but nevertheless performed remarkably given that x and y were hidden. Now, consider how these results change as increasing class noise is present. The left plot in Figure 10 compares the sensitivity of CCVD and COUNTS to class-noise as a function of the noise level (p percent of both training and test class labels are reassigned randomly from a uniform

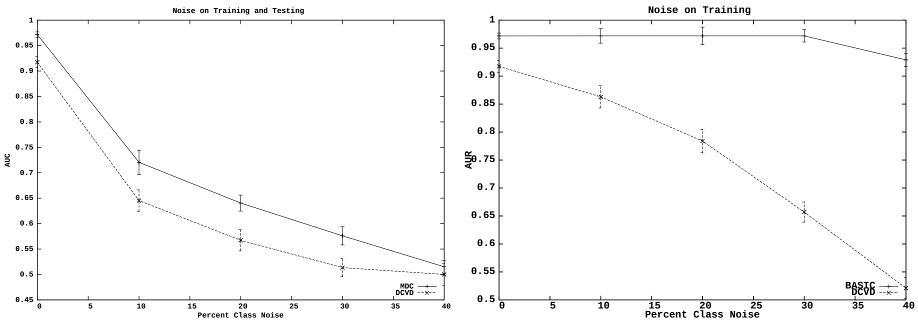

Fig. 10. Performance degradation for the AND domain as a function of the amount of 0/1 class noise. In the left plot both training and test sets were corrupted; the right plot shows results using a noisy training set and clean test set, as a measure of the ability to recover the true concept.

distribution). Both aggregation methods appear to be equally noise sensitive: the performance degradations track closely.

However, such a performance reduction has two components. First, the ability of the learner to recognize the underlying concept diminishes. Second, with increasing noise, the class labels in the test set are increasingly unpredictable. These effects can be separated by running the same experiment, except testing on uncorrupted (noise-free) data. The right plot of Figure 10 shows that COUNTS (provided x and y) indeed are able to learn the original concept with only minor degradation, despite up to 40% class noise. CCVD on the other hand shows a significant drop in performance (although somewhat less than before on the noisy test data). For COUNTS, even if 40% of the labels are potentially distorted, the other 60% still provide sufficient information to recognize the concept that larger counts of x and y are associated with positive class labels. The COUNTS aggregation can combine information about x and y from all bags and therefore is not very sensitive to the random variations.

On the other hand, for CCVD every bag contains information about a different set of identifiers. Each identifier appears only a few times, so the estimates of the class-conditional distributions are subject to significant variance errors. When using the identifiers as the predictors, noise in the class labels acts like noise in the predictors themselves; however, the x's and y's remain clean. In contrast, if *attribute* noise (misrecorded values of x and y) is present, we would expect the aggregates of identifier attributes to fare better. Indeed, Figure 11 shows that attribute noise affects only the COUNTS aggregates since CCVD does not use the noisy observations of x and y.

We have no firm basis to say which type of noise is more likely under what circumstances, but in cases where reliable attribute values are hard to get (e.g., because they are distorted, as with illegal activities) distribution-based aggregates can be a better choice. For example, for commercial fraud, it is often much

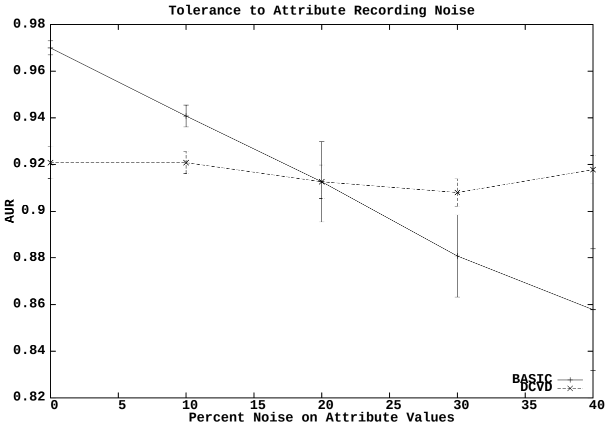

Fig. 11. Performance sensitivity on the AND domain to attribute recording noise for related entities. Since CCVD does not use the values of x and y (unobserved properties) it shows no performance decrease.

less costly to obscure attributes than to change identities frequently. Learning from identifiers does not require that the identity be true (e.g., that Peter Worthington is really Peter Worthington), but only that multiple actions can be related to the same person.

Domain	**1st**	**2nd**	**3rd**	**4th**	**5th**	**Min Appearance**
XOR 1	0.0082	0.0076	0.0076	0.0075	0.0075	35
XOR 2	0.1712	0.0754	0.0567	0.0567	0.0500	17
XOR 3	0.5533	0.1387	0.0942	0.0757	0.0705	8
XOR 4	0.9909	0.1859	0.01258	0.0945	0.0773	5

Table 9. Measures of the skewedness (differences in the likelihood of a entity to be related to some target case) of the relationship patterns: counts of the 5 most common values normalized by the number of target cases and the non-normalized count of the least common value. A uniform distribution (XOR 1) has low counts for the most common and a high count for the least common. As the skew increases (largest for XOR 4) the most common appearances increase and the least common decrease.

Relationship Patterns Another potential point of sensitivity of the distribution-based aggregation methods is the pattern of relationships among entities. For example, for AND and XOR, uniform distributions were used to assign related entities to target entities (each potentially related entity is equally likely to be chosen). In real-world domains (as we see in ours), it is often the case that the linkages are skewed—both that the degrees of nodes vary widely, and also that there is preferential attachment to particular entities (e.g., hubs on the Web).

To investigate sensitivity to skew, we simulate new versions of the XOR task with different skews of the relation distributions. Technically, skew (the third moment of a distribution) is only well defined for numeric distributions as a measure of symmetry. There is no symmetry for categorical distributions due to the lack of order. Thus, when we speak of a skewed relation distribution we mean that the probability of an entity to be related to some particular target case can differ significantly across entities. Unfortunately this cannot be quantified easily as in the numeric case of a third moment. Table 9 quantifies the skew of four different relation distributions in terms of the numbers of occurrences of the 5 most commonly related entities, normalized by the number of target objects (10000). The last column shows how often the least common value appeared. As the skew increases, the values for the 5 most common entities increase and the value of the least common appearance decreases. XOR1 represents the uniform distribution; XOR 4 is extremely skewed (99% of the target cases are linked to the most-common object). Table 10 compares the performances of the various aggregations on XOR1 and XOR4. For the strongly skewed data, earlier comparative conclusions remain the same with the exception of worse performance of the class-conditional distributions (CCVD), much better performance of the most discriminative values (MDC), and a strong relative improvement of combining the two. The performance of the combination is driven by the predictive information captured in MDC.

Domain	**COUNTS**	**MCC**	**UCVD**	**CCVD**	**MDC**	**MDC&CCVD**
XOR 1	0.50 (0.018)	0.51 (0.02)	0.62 (0.02)	0.92 (0.008)	0.51 (0.004)	0.92 (0.008)
XOR 4	0.51 (0.02)	0.49 (0.04)	0.71 (0.012)	0.78 (0.007)	0.75 (0.011)	0.86 (0.007)

Table 10. Ranking performance (AUC) on the XOR domain for uniform distribution (XOR 1) and highly skewed distribution (XOR 4), including standard deviations across 10 experiments.

The reason for the improvement of MDC is the large overlap of a few related entities. There are a few discriminative values (identifiers of particular objects with or without the XOR) that due to the skew appear in many training and generalization bags. For a uniform distribution, the class-conditional information for a particular value only provides information for a very small set of test cases that are also related to this value. The reduced performance of CCVD is a combination of two effects, the training size and the skew. Figure 12 shows the effects of the distribution of the related objects as a function of the skew (see Table 9) and the training size (XOR 1 is uniform and higher distributions have a stronger skew; see also the code in the Appendix were d is the skew parameter).

Observe the interesting pattern: for stronger skew, we see of better comparative performance for small training sizes, but (relatively) worse performance for large training sizes. The learning curves range from a steep gain for the no-skew

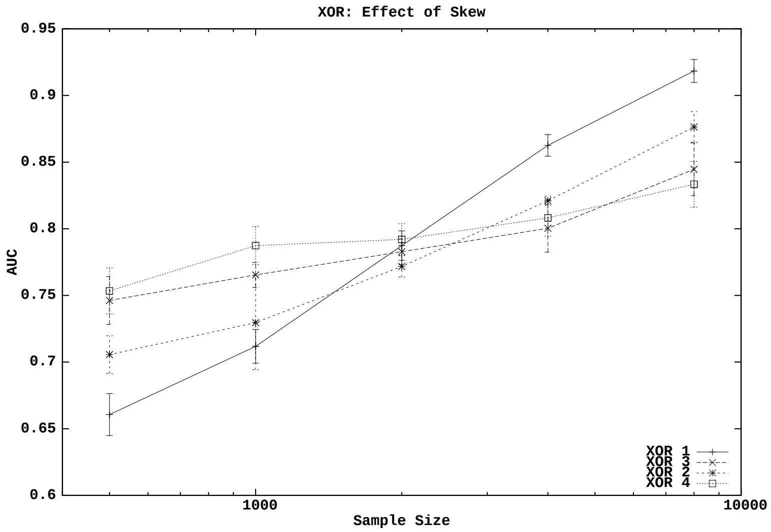

Fig. 12. Interaction effect of skew of relationship distribution and training size on ranking performance for the XOR domain. A stronger skew provides more useful information early, but the marginal value of additional training examples is lower.

uniform distribution to an almost flat learning curve for highly skewed relation distributions. The reason for this pattern is the difference in the amount of useful information available to the attribute construction process. With strong skew, even small training sets are sufficient to capture the information of the common related entities. This information is also very predictive for the test cases since they also are dominantly related to these same entities. However, as the training size increases little new information becomes available about the less-often related entities (because the skew works both ways). With enough training data, a uniform distribution provides in total more information because the marginal information for each additional training case is larger (cf., the Min Appearance column in Table 9). The relatively low performance (compared with the uniform case) for XOR4 of CCVD in Table 10 is a result of the large training size in combination with a high skew.

Domain Characterization The results in Table 7 use a large portion of the domain for training. The training size is of particular concern for aggregation based on distributional meta-data because of the large number of parameters to be estimated for the class-conditional distributions, and also because only part of the training data can be used for model induction and the rest must be reserved for estimating these parameters. The number of parameters is equal to the number of distinct values, for our domains: 10000 for XOR and AND, 490 for

KDD, 490 for IPO, 35000 for CORA, and 1860 for COOC. We now will examine generalization performance with very small training sets (250 examples).

Domain	1st	Min Appearance	Prior 1	Bag Size	AUC
Fraud	0.0005	1:666	0.01	20	0.48
XOR 1	0.0082	35	0.4	5	0.60
AND	0.0080	35	0.1	5	0.65
KDD	0.0609	1:14	0.06	3	0.74
IPO	0.1352	1:192	0.55	2	0.74
Cooc	0.183	1:616	0.27	26	0.78
Ebook	0.16	1:5854	0.06	28	0.84
CORA	0.0775	1:21460	0.32	20	0.90

Table 11. Performance (AUC) using cosine distance with small training sets (250 examples) as an interaction effect of skew (1st and Min Appearance, and where the latter equals 1 the number of values that appeared only once), unconditional prior of class 1, and average bag size.

Besides the amount of training data, there are various other characteristics of learning tasks that are important for assessing the applicability of different learning techniques, such as inherent discriminability, the number of features, the skew of the marginal class distribution (the class "prior"), and others [21],[22]. Relational domains have additional characteristics; particularly important in our case are two: the skew in the relationship distribution and the average size of bags of related values. We already have shown that a strong skew can improve performance with small training sets. The size of the bags determines the number of effective observations for the estimation of $P(t_{id}|c)$. Also directly important is the marginal class distribution, which determines the relative quality of the estimated positive and negative class-conditional distributions. For example, if only 1 percent of the target cases are positive, very few observations are available for $P(t_{id}|1)$ and many for $P(t_{id}|0)$; such class skew can be problematic if the minority class is much better defined ("customers who ...") than the majority class ("everyone else"), as is often the case.

Table 11 presents these three factors for all eight domains, and the ranking performance (AUC) with small training sets (250 training cases) using class-conditional cosine distances. The first two columns measure the skew: as the skew increases, the number of occurrences of the most commonly appearing value increases and the number of occurrences of the least common value decreases. For domains where the least common value appeared only once, we also include the number of distinct values that appear only once (after the colon). The table rows are ordered by increasing generalization performance.

We infer that the excellent performance on the CORA domain is a result of a relatively high prior (0.32), large bags (average of 20) and strong relation skew. Of the total of 35000 possible values, 21460 appear in only one bag—the estimate of $P(t_{id}|c)$ for these values therefore is irrelevant, and the effective number of parameters to be estimated is much lower than 35000. In particular the number of distinct values that appear in at least 10 bags is only 1169. The Ebook domain although having a much lower prior has good small-training-size performance due to a strong skew and large bags (in addition to a high inherent discriminability, as shown by the impressive results on the large training set in Table 7). AND and XOR suffer mostly from the uniform distribution of related objects as shown in Section 5.4 in addition to a small bag size. The lowest small-training-size performance is in the Fraud domain: the model does not provide any ranking ability at all. The reason is the combination of a very low prior of only 1 percent and a uniform distribution (by construction).

The upshot of these sensitivity analyses is a clarification of the conditions under which the attributes constructed based on vector distances to class-conditional distributions will be more or less effective. The class skew, the relationship skew, and the amount of training data affect whether there will be enough (effective) training cases to estimate the class-conditional distributions accurately. Additionally, the relationship skew determines how important it will be to estimate the class-conditional distributions well (in the presence of techniques like MDC, which get more effective with stronger relation skew).

5.5 Comparison to Other Relational Learners

We do not report a comprehensive study comparing ACORA to a wide variety of statistical relational modeling approaches (e.g., [23] [24] [25] [26]). This paper focuses on novel aggregation methods; ACORA is a vehicle for applying and studying these methods. We conjecture that these new aggregators ought to improve other relational learners as well. Indeed, except for the methods (such as PRMs) that include collective inferencing, ACORA is capable of approximating the other methods through appropriate choices of aggregators and model induction methods. They all follow a transformation approach that constructs features from the relational representation and then induces a propositional model from the new features. There are, of course, exceptions. For example, REGGLAGS [4] would be outside of ACORA's expressive power since it combines Boolean conditions and aggregation and can form more complex aggregations (cf., Perlich and Provost's hierarchy of aggregation complexity [2]).

More importantly, the domains used in this paper (with the exception of IPO and EBooks) simply are not suitable for any of the above systems. To our knowledge, none has the ability to aggregate high-dimensional categorical attributes automatcally, and without those attributes only propositional data and known class labels remain.

It is possible to compare classification accuracy with logic-based systems such as FOIL, but the problem remains: such systems require the identification of constants that may be used for equality tests in the model. Without the identifier attributes, they also have no information except for the few attributes in EBook and IPO. To illustrate, we compare (on the IPO domain) ACORA to four logic-based relational learners (FOIL [27], TILDE [28], Lime [29], and Progol [30]). Since ILP systems typically (with the exception of TILDE) only predict the class, not the probability of class membership, we compare in Table 12 the accuracy as a function of training size. We also include as a reference point the classification performance of a propositional logistic model without any background knowledge (NO). ACORA uses a stopping criteria of depth = 3 and logistic regression for model induction.

We selected four ILP methods: FOIL [27] uses a top-down, separate-and-conquer strategy adding literals to the originally empty clause until a minimum accuracy is achieved. TILDE [28] learns a relational decision tree using FOL clauses in the nodes to split the data. Lime [29] is a top-down ILP system that uses Bayesian criteria to select literals. Progol [30] learns a set of clauses following a bottom-up approach that generalizes the training examples. We did not provide any additional (intentional) background knowledge beyond the facts in the database. We supplied declarative language bias for TILDE, Lime, and Progol (as required). For these results, the banks were not allowed as model constants.

Size	**NO**	**FOIL**	**TILDE**	**Lime**	**Progol**	**CCVD**
250	0.649	0.645	0.646	0.568	0.594	0.713
500	0.650	0.664	0.628	0.563	0.558	0.78
1000	0.662	0.658	0.630	0.530	0.530	0.79
2000	0.681	0.671	0.650	0.512	0.541	0.79

Table 12. Accuracy comparison with logic-based relational classifiers (FOIL, Tilde, Lime, Progol), target features (TF), and using no relational information (NO) as a function of training size.

The results in Table 12 demonstrate that the logic-based systems simply are not applicable to this domain. The class-conditional distribution features (CCVD) improve substantially over using no relational information at all (NO), so there indeed is important relational information to consider. The ILP systems FOIL and TILDE never perform significantly better than using no relational information, and Progol and Lime often do substantially worse.

Given that we excluded banks from the permissible constraints for equality tests, there was no attribute in the related objects that any of the ILP methods could have used. Allowing all constants including identifiers to be used for equality tests is similar to constructing count aggregates for all values. However, given the extreme increase in run times we were only able to run this experiment using TILDE. Since TILDE is able to predict probabilities using the class frequencies at the leaves, we can compare (in Table 13) its AUC to our earlier results.[10] Based on these results we must conclude that except for the EBook and the IPO domain, TILDE could not generalize a classification model from the provided identifier attributes. We conjecture that TILDE can only take advantage

[10] On the IPO domain TILDE improved also in terms of accuracy over the performance without banks in Table 12 from 0.65 to 0.753.

Domain	COUNTS	Tilde	CCVD	MCC
XOR	0.5 *	0.5 (0)	0.92 (0.008)	0.51 (0.004)
AND	0.5 *	0.5 (0)	0.92 (0.006)	0.52 (0.012)
Kohavi	0.5*	0.5 (0)	0.85 (0.025)	0.71 (0.022)
IPO	0.70* (0.023)	0.76 (0.28)	0.79 (0.03)	0.77 (0.02)
CORA	0.5*	0.5 (0)	0.97 (0.003)	0.74 (0.018)
COOC	0.5*	0.5 (0)	0.78 (0.02)	0.63 (0.016)
EBook	0.716 (0.024)	0.83 (0)	0.95 (0.024)	0.79 (0.011)
Fraud	0.5*	0.5 (0)	0.87 (0.028)	0.49 (0.005)

Table 13. Comparison of generalization performance (AUC) for different aggregation strategies (see Table 6 for a description). Entries with * denote cases where the COUNTS aggregation was not applicable because all categorical attributes had too many distinct values. The standard deviation across 10 experiments is included in parenthesis.

of a strong and concentrated signal. Both domains IPO and EBook also show relatively good performance of MCC. This suggests that there are a few identifier values that are both predictive and relatively frequent. If the discriminative power of a particular value or its frequency was too low, TILDE did not use it. This highlights again that the ability to condense information across multiple identifier values is necessary to learn predictive models.

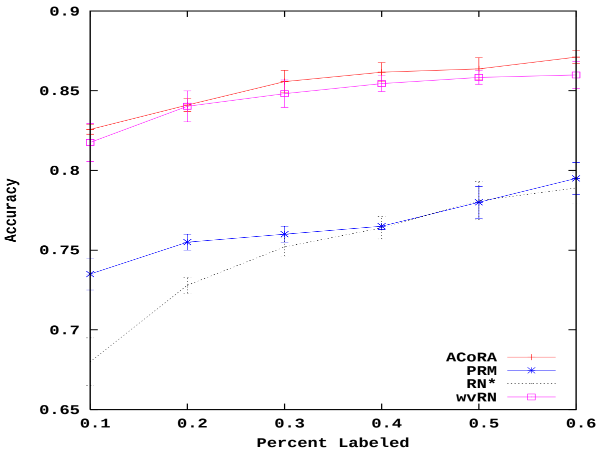

Fig. 13. Comparison of classification accuracy of ACORA using class-conditional distributions against a Probabilistic Relational Model (PRM) and a Simple Relational Classifier (SRC) on the CORA domain as a function of training size.

Figure 13 shows that using identifier attributes would likely have improved other published relational learning results as well. For the Cora domain, the figure shows classification accuracies as a function of training size. ACORA estimates 7 separate binary classification models using class-conditional distributions for each of the 7 classes and predicts the final class with the highest probability score across the 7 model predictions. The figure compares ACORA to prior published results using Probabilistic Relational Models (PRM, [23]) based on both text and relational information (as reported by [31]), and a Simple Relational Classifier (SRC, [10]) that assumes strong autocorrelation in the class labels (specifically, assuming that documents from a particular field will dominantly cite previously published papers in the same field), and uses relaxation labeling to estimate unknown classes. Again ACORA using identifier attributes (the particular papers) and target features dominates the comparison, even for very small training sets. The main advantage that ACORA has over the PRM is the ability to extract information from the identifier attributes of authors and papers. The PRM uses the identifiers to construct its skeleton, but does not include them explicitly (does not estimate their distributions) in the model.

6 Prior and Related Work

There has been no focused work within relational learning on the role of identifiers as information carriers. There are three main reasons: 1) a historical reluctance within propositional learning to use them because they cannot generalize; 2) the huge parameter space implied by using identifiers as conventional categorical values, which typically is not supported by sufficient data (potentially leading to overfitting and excessive run time), and 3) the commonly assumed objective of making predictions in a "different world" where none of the training objects exist, but only objects with similar attributes.

In contrast to a large body of work on model estimation and the estimation of functional dependencies that map well-defined input spaces onto output spaces, aggregation operators are much less well investigated. Model estimation tasks are usually framed as search over a structured (either in terms of parameters or increasing complexity) space of many possible solutions. Although aggregation has been identified as a fundamental problem for relational learning from real-world data [32], machine learning research has considered only a limited set of aggregation operators. Furthermore, statistical relational model estimation typically treats aggregation as a preprocessing step that is independent of the model estimation process. In Inductive Logic Programming ([33]) and logic-based propositionalization, aggregation of one-to-many relationships is achieved through existential quantification and is part of the active search through the model space.

Propositionalization ([1], [4], [19]) has long recognized the essential role of aggregation in relational modeling. This work focuses specifically on the effect of aggregation choices and parameters, yielding promising empirical results on noisy real-world domains: the numeric aggregates in [1] outperform three ILP systems

(FOIL [27], Tilde [28], and Progol [30]) on a noisy financial task (PKDD-CUP 2000). Krogel and Wrobel ([4], [19]) show similar results on the financial task and a customer-classification problem (ECML 1998 discovery challenge) in comparison to Progol and Dinus [34], a logic-based propositionalization approach. Similar work by Krogel et al. [35] presents an empirical comparison of Boolean and numeric aggregation in propositionalization approaches across multiple domains, including synthetic and domains with low noise; however their results are inconclusive. Perlich and Provost [2] find that logic-based relational learning and logic-based propositionalization perform poorly on a noisy domain compared to numeric aggregation. They also discuss theoretically the implications of various assumptions and aggregation choices on the expressive power of resulting classification models and show empirically 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.

Distance-based relational approaches [36] use simple aggregates such as MIN to aggregate distances between two bags of values. A first step estimates the distances between all possible pairs of objects (one element from each bag) and a second step aggregates all distances through MIN. The recent convergence of relational learning and kernel methods has produced a variety of kernels for structured data, see for instance [37]. Structured kernels estimate distances between complex objects and are typically tailored towards a particular domain. This distance estimation also involves aggregation and often uses sums.

Statistical relational learning approaches [38] [3] include network models as well as upgrades of propositional models (e.g., Probabilistic Relational Models [23], Relational Bayesian Classifier [24], Relational Probability Trees [25]). They typically draw from a set of simple numeric aggregation operators (MIN, MAX, SUM, $MEAN$ for numerical attributes and $MODE$ and $COUNTS$ for categorical attributes with few possible values) or aggregate by creating Boolean features (e.g., Structural Logistic Regression [26], Naive Bayes with ILP [39]). Krogel and Wrobel [4] and Knobbe et al. [1] were to our knowledge the first to suggest the combination of such numerical aggregates and FOL clauses to propositionalize relational problems automatically.

Besides special purpose methods (e.g., recency and frequency for direct marketing) only a few new aggregation-based feature construction methods have been proposed. Craven and Slattery [40] use Naive Bayes in combination with FOIL to construct features for hypertext classification. Perlich and Provost [2] use vector distances and class-conditional distributions for noisy relational domains with high-dimensional categorical attributes. (This paper describes an extension of that work.) Flach and Lachiche ([15],[16]) develop a general Bayesian framework that is closely related to our analysis in Section 4.2 but apply it only to normal attributes with limited cardinality.

Theoretical work outside of relational modeling investigates the extension of relational algebra [41] through aggregation; however it does not suggest new operators. Libkin and Wong [42] analyze the expressive power of relational languages with bag aggregates, based on a count operator and Boolean comparison (sufficient to express the common aggregates like $MODE$ and MAX). This might prove to be an interesting starting point for theoretical work on the expressiveness of relational models.

Traditional work on constructive induction (CI) [43] stressed the importance of the relationship between induction and representation and the intertwined search for a good representation. CI focused initially on the capability of "formulating new descriptors" from a given set of original attributes using general or domain-specific constructive operators like $AND, OR, MINUS, DIVIDE$, etc. Wnek and Michalski [44] extended the definition of CI to include any change in the representation space while still focusing on propositional reformulations. Under the new definition, propositionalization and aggregation can be seen as CI for relational domains as pointed out by [45,46] and [47] for logic-based approaches.

7 Conclusion

We have presented novel aggregation techniques for relational classification, which estimate class-conditional distributions to construct discriminative features from relational background tables. The main technique uses vector distances for dimensionality reduction and is capable of aggregating high-dimensional categorical attributes that traditionally have posed a significant challenge in relational modeling. It is implemented in a general relational learning prototype ACORA that is applicable to a large class of relational domains with important information in identifier attributes, for which the traditional $MODE$ aggregator is inadequate.

The main theoretical contributions of this work are the analysis of desirable properties of aggregation operators for predictive modeling, the derivation of a new aggregation approach based on distributional meta-data for a "relational fixed-effect" model, and the exploration of opportunities, such as learning from unobserved object characteristics, arising from the aggregation of object identifiers. The commonly made assumption of class-conditional independence of attributes significantly limits the expressive power of relational models and we show that the aggregation of identifiers can overcome such limitations. The ability to account for high-dimensional attributes encourages the explicit exploration of attribute dependencies through the combination of values from multiple attributes—which we have not explored here, but is an important topic for future work. We also conduct a comprehensive empirical study of aggregation for identifier attributes. The results demonstrate that the new approach indeed allows generalization from identifier information, where prior aggregation approaches fail. This is due primarily to the ability of the new aggregation operators to reduce dimensionality while constructing discriminative features that exhibit task-specific similarity, grouping cases of the same class together. Our results also support claims that learning from identifiers allows the capture of concepts based on unobserved attributes and concepts that violate the assumptions of class-conditional independence, and can reduce the need for deep exploration of the network of related objects (even if the "true" concept is based on deeper relationships). By using real and synthetic data sets with different characteristics, we illustrate the interactions of training size, marginal class distribution, average number of related objects, and the degree of skew in the distribution of related objects. For example, for small data sets higher skew, larger bags, and more balanced class priors increase the generalization performance of the class-conditional distribution-based aggregates.

We also introduce an aggregation method using counts of most-discriminative values (also based on the distributional meta-data), which generally outperforms counts of most common values, and in particular profits from a high skew of related objects. Discriminative counts are typically not as predictive as the distances between the class-conditional distributions, but provide additional predictive power in domains with high skew.

The distribution-based approach to aggregation is not limited to categorical values. Via discretization it can also be applied to numeric attributes with arbitrary distributions. Define the density function of a numeric attribute as the derivative of the cumulative density function $F(X)$ (the probability of observing an $x \leq X$). As usual, the derivative is the limit of h going to infinity of $(F(X+h)-F(x))/h$. Let h be the bin size for discretization. As the number of training cases increases the proportion of elements falling into bins below X will converge to $F(X)$. Letting the bin size h go to 0 reaches in the limit the density function of an arbitrary numeric distribution.

The view of feature construction as computing and storing distributional meta-data allows the application of the same idea to regression tasks or even unsupervised problems. For instance, it is possible to find clusters of all (related) objects and define a cluster (rather than a class-conditional distribution) as the reference point for feature construction.

Finally, this work highlights the sensitivity of generalization performance of relational learning to the choice of aggregators. We hope that this work provides some motivation for further exploration and development of aggregation methods for relational modeling tasks.

Acknowledgments

We are grateful to William Greene, David Jensen, Sofus Macskassy, Ted Senator, and Jeff Simonoff and our anonymous reviewers for their valuable comments. David Jensen suggested viewing our reference distributions as meta-data. We also wish to thank Jae Yun Moon and her husband for providing the EBook database. Finally we are gratful to Hendrick Bloeckeel and Jan Struyf for providing us with the newest version of TILDE and to include a special option to output probability estimates.

This research was sponsored in part by the Air Force Research Laboratory, Air Force Materiel Command, USAF, under Agreement number F30602-01-2-0585. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL or the U.S. Government.

Appendix A: Domain Description

Below are brief descriptions of the domains used for the empirical evaluations. The table gives summary statistics on the number of numeric, categorical (with fewer than 100 possible values), and identifier attributes (categoricals with more than 100 distinct values). The target table appears in bold.

XOR and AND

Each domain comprises two tables: target objects o and related entities e. Related entities have three fields: an identifier and two **unobserved** Boolean fields x and y that are randomly assigned (uniformly). Each target object is related to k entities; k is drawn from a uniform distribution between 1 and upper bound u. The expected value of k is therefore $(u + 1)/2$ and is 5 in our main comparison. The likelihood that an entity is related to a target object is a function of its identifier number. For the main comparison this is also uniform. Followup experiments (in Section 5.4) will vary both k and the distributions of related entities. For XOR the class of a target object is 1 if and only if the XOR between x and y is true for the majority of related entities. XOR represents an example of a task where the aggregation of x and y independently (i.e., assuming class-conditional independence) cannot provide any information. However, the identifiers have the potential to proxy for the entities' XOR values. For AND the class of a target object is 1 if and only if the majority of related entities satisfy $x = 1$ AND $y = 1$. This concept also violates the independence assumption. However, aggregations of bags of x's or y's using counts can still be predictive. To demonstrate the ability of learning from unobserved attributes we do not include in the main results the values of x and y but provide only the identifier.

Code for the generation of related entities with identifier ”oid” and the attributes x,y for the calculation of the class label:

```
$num=$ARGV[0];
open OUT, ">objects.rel"
$i=1;
while($i<=$num)
{
    $x=rand();
    if ($x<0.5)\{$x=0}else\{$x=1};
    $y=rand();
    if ($y<0.5)\{$y=0}else\{$y=1};
    print OUT "oid$i\n";
    #print OUT "oid$i,$x,$y\n";
    $i++;
}
close OUT;
```

Code for the generation of the target object and the relationships: The parameter $rel is the average number of related entities whereas $d regulates the skew of the likelihood that an entity is chosen.

```
$stem=$ARGV[0];
$rel=$ARGV[1];
$d=$ARGV[2];

$count=10000;

open IN, "objects.rel";
@o=();

while($in=<IN>)
{
    chop $in;
    push @o, $in;
}

$i=0;
open TAR,">$stem"."_tar.rel" or die;
open REL,">$stem"."_rel.rel";
while($i<=$count)
{
    $tar=0;
    $c=int rand()*2*$rel+1;
    $cc=$c;
    while($c>=1)
    {
$v=int rand()**$d*$#o;
($b,$x,$y,$z)=split /,/, $o[$v];
$tar+=$z;
print REL "tar$i,$b\n";
$c+=-1;
    }
    $res=0;
    if ($tar/$cc>0.5)\{$res=1}
    print TAR "tar$i,$res\n";
    $i++;
}
```

Synthetic Telephone Fraud

This synthetic domain isolates a typical property of a telephone network with fraudulent use of accounts. The only objects are accounts, of which a small fraction (1 %) are fraudulent. These fraudulent accounts have the property of making a (larger than usual) proportion of their calls to a set F of particular (non-fraudulent) accounts. This is the basis of one type of highly effective fraud-detection strategy [13][18]; there are many variants, but generally speaking accounts are flagged as suspicious if they call numbers in F. The code generates a set of 1000 fraudulent accounts and 99000 normal accounts. Normal users call other accounts randomly with a uniform distribution over all accounts. Fraudulent users make 50% of their calls to a particular set of numbers (1000 numbers that are not fraudulent accounts) with uniform probability of being called, and 50% randomly to all accounts.

```
# number of accounts: 100000
# fraud accounts are 99001 to 10000
# fraud numbers are 1:1000

open TAR,">fraud.rel";
open REL,">calls.rel";
$i=1;
while($i<=100000){
    $tar=0;
    $c=int rand()*30+1;    #average number of calls =15
    if ($i>99000)          #1:1000 is fraud account
    {print TAR "n$i,1\n";}
    else{print TAR "n$i,0\n";}
    while($c>=1){           #generate calls
        if ($i>99000)}      #fraud accoun
    if (rand()<0.5){$num=rand()*1000} #fraud number
    else{$num=rand()*10000}}           #not fraud
        else{               # normal account
    if (rand()<0.25){$num=10000+rand()*99000;}
    else{$num=rand()*100000}}
        $num=int $num;
        print REL "n$i,n$num\n";
        $c=$c-1;}
    $i=$i+1;}
close TAR;
close REL;
```

Customer Behavior (KDD)

Blue Martini [48] published, together with the data for the KDDCUP 2000, three additional customer data sets to evaluate the performance of association rule algorithms. We use the BMS-WebView-1 set of 59600 transactions with 497 distinct items. The classification task is the identification of transactions that contained the most commonly bought item (12895), given all other items in the transaction.

Direct Marketing (EBooks)

Ebooks comprises data from a five-year-old Korean startup that sells E-Books. The database contains many tables; we focus on the customer table (attributes include, for example, country, gender, mailing preferences, and household information) and the transaction table (price, category, and identifier). The classification task is the identification of customers that bought the most commonly bought book (0107030800), given all other previously bought items.

Industry Classification (COOC)

This domain is based on a corpus of 22,170 business news stories from the 4-month period of 4/1/1999 to 8/4/1999, including press releases, earnings reports, stock market news, and general business news [49]. For each news story there is a set of ticker symbols of mentioned firms, which form a co-occurrence relation between pairs of firms. The classification task is to identify Technology firms, labeled according to Yahoo's industry classification (table T), given their story co-occurrences with other firms (table C).

Initial Public Offerings (IPO)

Initial Public Offerings of firms are typically headed by one bank (or occasionally multiple banks). The primary bank is supported by a number of additional banks as underwriters. The job of the primary bank is to put shares on the market, to set a price, and to guarantee with its experience and reputation that the stock of the issuing firm is indeed valued correctly. The IPO domain contains three tables, one for the firm going public, one for the primary bank, and one for underwriting banks. Firms have a number of numerical and categorical attributes but for banks only the name is available. The classification task is to predict whether the offer was (would be) made on the NASDAQ exchange.

Document Classification (CORA)

The CORA database [9] contains 4200 publications in the field of Machine Learning that are categorized into 7 classes: Rule Learning, Reinforcement Learning, Theory, Neural Networks, Probabilistic Methods, Genetic Algorithms, and Case-Based Reasoning. We use only the authorship and citation information (without the text) as shown previously in Figure 4. We focus for the main results only on the most prevalent class: Neural Networks. The full classification performance using the maximum probability score across all 7 classes can be found later in Figure 13.

References

1. Knobbe, A., Haas, M.D., Siebes, A.: Propositionalisation and aggregates. In: Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD). (2001) 277–288
2. Perlich, C., Provost, F.: Aggregation-based feature invention and relational concept classes. In: Proceedings of the Ninth ACM International Conference on Knowledge Discovery and Data Mining (KDD). (2003)
3. Jensen, D., (editors), L.G.: Proceedings of the Workshop on Learning Statistical Models from Relational Data (IJCAI). (2003)
4. Krogel, M.A., Wrobel, S.: Transformation-based learning using multirelational aggregation. In: Proceedings of the 11th International Conference on Inductive Logic Programming (ILP). (2001)
5. Woznica, A., Kalousis, A., Hilario, M.: Kernel-based distances for relational learning. In: Proceedings of the Workshop on Multi-Relational Data Mining (KDD). (2004)
6. Gärtner, T., Lloyd, J.W., Flach, P.A.: Kernels for structured data. In: Proceedings of the 12th International Conference on Inductive Logic Programming (ILP). (2002)
7. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, Los Altos, California (1993)
8. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann (1999)
9. McCallum, A., Nigam, K., J.Rennie, Seymore, K.: Automating the construction of internet portals with machine learning. Information Retrival **3** (2000) 127–163
10. Macskassy, S., Provost, F.: A simple relational classifier. In: Proceedings of the Workshop on Multi-Relational Data Mining (KDD). (2003)
11. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of the 7th ACM International Conference on Knowledge Discovery and Data Mining (KDD). (2001) 57–66
12. Chakrabarti, S., Dom, B., Indyk, P.: Enhanced hypertext categorization using hyperlinks. In: Proceedings of the ACM International Conference on Management of Data. (1998) 307–318
13. Fawcett, T., Provost, F.: Adaptive fraud detection. Data Mining and Knowledge Discovery **1** (1997) 291–316
14. DerSimonian, R., Laird, N.: Meta-analysis in clinical trials. Controlled Clinical Trials **7** (1986) 177 – 188
15. Flach, P., Lachiche, N.: Naive Bayesian classification for structured data. Machine Learning (2004) 233–269
16. Lachiche, N., Flach, P.A.: 1bc2: A true first-order bayesian classifier. In: "Proceedings of the 12th International Conference on Inductive Logic Programming (ILP). (2002) 133–148
17. Jensen, D., Neville, J.: Linkage and autocorrelation cause feature selection bias in relational learning. In: Proceedings of the 19th International Conference on Machine Learning (ICML). (2002)

18. Cortes, C., Pregibon, D., Volinsky, C.: Communities of interest. Intelligent Data Analysis **6(3)** (2002) 211–219
19. Krogel, M.A., Wrobel, S.: Facets of aggregation approaches to propositionalization. In: Proceedings of the 13th International Conference on Inductive Logic Programming (ILP). (2003) 30–39
20. Bradley, A.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition **30(7)** (1997) 1145–1159
21. Brazdil, P., Gama, J., Henery, R.: Characterizing the applicability of classification algorithms using meta level learning. In: Proceedings of the 7th European Conference on Machine Learning (ECML). (1994) 83–102
22. Perlich, C., Provost, F., Simonoff, J.: Tree induction vs. logistic regression: A learning-curve analysis. Journal of Machine Learning Research **4** (2003) 211–255
23. Koller, D., Pfeffer, A.: Probabilistic frame-based systems. In: Proceedings of the 15th National Conference on Artificial Intelligence (AAAI). (1998) 580–587
24. Neville, J., Jensen, D., Gallagher, B., Fairgrieve, R.: Simple estimators for relational bayesian classifiers. Technical report, 03-04, University of Massachusetts (2003)
25. Jensen, D., Neville, J.: Data mining in social networks. In: Procedings of the National Academy of Sciences Symposium on Dynamic Social Networks Modeling and Analysis. (2002)
26. Popescul, A., Ungar, L.H., Lawrence, S., Pennock, D.M.: Structural logistic regression: Combining relational and statistical learning. In: Proceedings of the Workshop on Multi-Relational Data Mining (KDD). (2002) 130–141
27. Quinlan, J., Cameron-Jones, R.: FOIL: A midterm report. In: Proceedings of the 6th European Conference on Machine Learning (ECML). (1993) 3–20
28. Blockeel, H., Raedt, L.D.: Top-down induction of first-order logical decision trees. Artificial Intelligence **101** (1998) 285–297
29. McCreath, E.: Induction in First Order Logic from Noisy Training Examples and Fixed Example Set Size. PhD thesis, Universtity of New South Wales (1999)
30. Muggleton, S.: CProgol4.4: a tutorial introduction. In Dzeroski, S., Lavrac, N., eds.: Relational Data Mining, Springer-Verlag (2001) 105–139
31. Taskar, B., Segal, E., Koller, D.: Probabilistic classification and clustering in relational data. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI). (2001) 870–878
32. Goldberg, H., Senator, T.: Restructuring databases for knowledge discovery by consolidation and link formation. In: Proceedings of the 1st International Conference On Knowledge Discovery and Data Mining (KDD). (1995)
33. Muggleton, S., DeRaedt, L.: Inductive logic programming: Theory and methods. The Journal of Logic Programming **19 & 20** (1994) 629–680
34. Lavrac, N., Dzeroski, S.: Inductive Logic Programming: Techniques and Application. Ellis Horwood, New York (1994)
35. Krogel, M.A., Rawles, S., Železný, F., Flach, P., Lavrač, N., Wrobel, S.: Comparative evaluation of approaches to propositionalization. In: 13th International Conference on Inductive Logic Programming (ILP). (2003) 197–214
36. Kirsten, M., Wrobel, S., Horvath, T.: Distance based approaches to relational learning and clustering. In Dzeroski, S., Lavrac, N., eds.: Relational Data Mining. Springer Verlag (2000) 213–232
37. Gärtner, T.: A survey of kernels for structured data. SIGKDD Explorations **5** (2003) 49–58

38. Neville, J., Rattigan, M., Jensen, D.: Statistical relational learning: Four claims and a survey. In: Proceedings of the Workshop on Learning Statistical Models from Relational Data (IJCAI). (2003)
39. Pompe, U., Kononenko, I.: Naive bayesian classifier with ilp-r. In: Proceedings of the 5th International Workshop on Inductive Logic Programming. (1995) 417–436
40. Craven, M., Slattery, S.: Relational learning with statistical predicate invention: Better models for hypertext. Machine Learning **43** (2001) 97–119
41. Özsoyoğlu, G., Özsoyoğlu, Z., Matos, V.: Extending relational algebra andrelational calculus with set-valued atributes and aggregate functions. In: ACM Transactions on Database Systems. Volume 12. (1987) 566–592
42. Libkin, L., L.Wong: New techniques for studying set languages, bag languages and aggregate functions. In: Proceedings of the thirteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. (1994) 155 – 166
43. Michalski, R.: A theory and methodology of inductive learning. Artificial Intelligence **20** (1983) 111–161
44. Wnek, J., Michalski, R.: Hypothesis-driven constructive induction in aq17-hci: A method and experiments. **14** (1993) 139–168
45. Kietz, J.U., Morik, K.: A polynomial approach to the constructive induction of structural knowledge. Machine Learning **14** (1994) 193 – 217
46. Morik, K.: Tailoring representations to different requirements. In: Proceedings of the 10th International Conference on Algorithmic Learning Theory (ALT). (1999) 1–12
47. Kramer, S., Lavrac, N., Flach, P.: Propositionalization approaches to relational data mining. In Dzeroski, S., Lavrac, N., eds.: Relational Data Mining. Springer-Verlag (2001) 262–291
48. Zheng, Z., Kohavi, R., Mason, L.: Real world performance of association rule algorithms. In: Proceedings of the 7th ACM International Conference on Knowledge Discovery and Data Mining (KDD). (2001) 401–406
49. Bernstein, A., Clearwater, S., Hill, S., Perlich, C., Provost, F.: Discovering knowledge from relational data extracted from business news. In: Proceedings of the Workshop on Multi-Relational Data Mining (KDD). (2002)