Mach Learn (2006) 62: 65–105
DOI 10.1007/s10994-006-6064-1

Distribution-based aggregation for relational learning with identifier attributes

Claudia Perlich · Foster Provost

Received: 11 August 2004 / Revised: 22 February 2005 / Accepted: 05 July 2005 /
Published online: 27 January 2006

Abstract Identifier attributes—very high-dimensional categorical attributes such as particular product ids or people's names—rarely are incorporated in statistical modeling. However, they can play an important role in relational modeling: it may be informative to have communicated with a particular set of people or to have purchased a particular set of products. A key limitation of existing relational modeling techniques is how they aggregate bags (multisets) of values from related entities. The aggregations used by existing methods are simple summaries of the distributions of features of related entities: e.g., MEAN, MODE, SUM, or COUNT. This paper's main contribution is the introduction of aggregation operators that capture more information about the value distributions, by storing meta-data about value distributions and referencing this meta-data when aggregating—for example by computing class-conditional distributional distances. Such aggregations are particularly important for aggregating values from high-dimensional categorical attributes, for which the simple aggregates provide little information. In the first half of the paper we provide general guidelines for designing aggregation operators, introduce the new aggregators in the context of the relational learning system ACORA (Automated Construction of Relational Attributes), and provide theoretical justification. We also conjecture special properties of identifier attributes, e.g., they proxy for unobserved attributes and for information deeper in the relationship network. In the second half of the paper we provide extensive empirical evidence that the distribution-based aggregators indeed do facilitate modeling with high-dimensional categorical attributes, and in support of the aforementioned conjectures.

Keywords: identifiers · relational learning · aggregation · networks

Editors: Hendrik Blockeel, David Jensen and Stefan Kramer

C. Perlich (✉)
IBM T.J. Watson Research Center
e-mail: perlich@us.ibm.com

F. Provost
New York University
e-mail: fprovost@stern.nyu.edu

1. Introduction

Predictive modeling often is faced with data including important relationships between entities. For example, customers engage in transactions which involve products; suspicious people may make phone calls to the same numbers as other suspicious people. Extending the traditional "propositional" modeling approaches to account for such relationships introduces a variety of opportunities and challenges. The focus of this paper is one such challenge—the integration of information from one-to-many and many-to-many relationships: a customer may have purchased many products; a person may have called many numbers.

Such n-to-many relationships associate with any particular entity a **bag** (multiset) of related entities. Since the ultimate objective of much predictive modeling is to estimate a single value for a particular quantity of interest, the predictive model must either ignore the bags of related entities or **aggregate** information from them.

The aggregation operators used by existing relational modeling approaches typically are simple summaries of the distributions of features of related entities, e.g., MEAN, MODE, SUM, or COUNT. These operators may be adequate for some features, but fail miserably for others. In particular, if the bag consists of values from high-dimensional categorical attributes, simple aggregates provide little information. **Object identifiers** are one instance of high-dimensional categorical attributes, and they are abundant in relational domains since they are necessary to express the relationships between objects. Traditional propositional modeling rarely incorporates object identifiers, because they typically hinder generalization (for example by creating "lookup tables"). However, the identities of *related* entities can play an important role in relational modeling: it may be informative to have communicated with a specific set of people or to have purchased a specific set of products. For example, Fawcett and Provost (1997) show that incorporating particular called-numbers, location identifiers, merchant identifiers, etc., can be quite useful for fraud detection.

Consider the following example of a simple relational domain that exhibits such n-to-many relationships. The domain consists of two tables in a multi-relational database: a **target table**, which contains one row for each of a set of **target entities**, about which some attribute value will be estimated, and an auxiliary table that contains multiple rows of additional information about entities related to the target entities. Figure 1 illustrates the case of a customer table and a transaction table. This simple case is ubiquitous in business applications, such as customer classification for churn management, direct marketing, fraud detection, etc. In each, it is important to consider transaction information such as types, amounts, times, and locations. Traditionally practitioners have manually constructed features before applying a conventional propositional modeling technique such as logistic regression. This manual process is time consuming, becomes infeasible for large and complex domains, and rarely will provide novel and surprising insights.

Relational learning methods address the need for more automation and support of modeling in such domains, including the ability to explore information about the many-to-many relationship between customers and products. If the modeling objective is to estimate the likelihood of responding to an offer for a particular book, it may be valuable to incorporate the specific books previously bought by the customer, as captured by their ISBNs. The MODE clearly is not suitable to aggregate a bag of ISBNs, since typically books are bought only once by a particular customer. In addition, this MODE feature would have an extremely large number of possible values, perhaps far exceeding the number of training examples.

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(CID, TYPE, ISBN, PRICE)

We introduce novel aggregators[1] that allow learning techniques to to capture information from identifiers such as ISBNs. This ability is based on (1) the implicit reduction of the dimensionality by making (restrictive) assumptions about the number of distributions from which the values were generated, and (2) the use of *distances* to class-conditional, distributional meta-data. Such distances reduce the dimensionality of the model estimation problem while maintaining discriminability among instances, and they focus explicitly on discriminative information.

The contributions of this work include:

1. An analysis of principles for developing new aggregation operators (Section 2).
2. The development of a novel method for relational feature construction, based on the foregoing analysis, which includes novel aggregation operators (Section 3). To our knowledge, this is the first relational aggregation approach that can be applied generally to categorical attributes with high cardinality.
3. A theoretical justification of the approach that draws an analogy to the statistical distinction between random- and fixed-effect modeling, and identifies typical aggregation assumptions that limit the expressive power of relational models (Section 3.4).
4. A theoretical conjecture (Section 3.5) that the aggregation of identifier attributes can implicitly support the learning of models from *unobserved* object properties.
5. An extensive empirical study demonstrating that the novel aggregators indeed can improve predictive modeling in domains with important high-dimensional categorical attributes, including a sensitivity analysis of major domain properties (Section 4).

The proposed aggregation methodology can be applied to construct features from various attribute types and for a variety of modeling tasks. We will focus in this paper on high-dimensional categorical attributes, and on classification and the estimation of class-membership probabilities. Unless otherwise specified we will assume binary classification.

[1] This paper is an extension of the second half of a prior conference paper Perlich and Provost (2003).

2. Design principles for aggregation operators

Before we derive a new aggregation approach for categorical attributes with high cardinality, let us explore the objectives and some 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 far removed from the estimation of generalization performance (e.g., while following a chain of relations). In addition, aggregation almost always involves 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 possible unique bags. For example for the variable TYPE in our example, there are fifty-four possible unique, non-empty bags with size less than ten containing values from $\{$Fiction, Non-Fiction$\}$. Consider two simple aggregation operators: *MODE* and *COUNT*. *MODE* has two possible aggregate values and *COUNT* has nine. Both lose considerable information about the content of the bags, and one might 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 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.*

[2] Related issues of quantifying the goodness of transformation operators have been raised by Gärtner et al.(2002) in the context of "good kernels" for structured data.

Thus, we face a tradeoff between instance discriminability and similarity preservation. Coding maintains instance discriminability perfectly, but obscures almost certainly the similarity. *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. However, whether or not *COUNT* or *MODE* are predictive will depend on the modeling task. They do not induce a task-specific similarity as their values are independent of the particular class labels.

Furthermore, since most similarity-preserving operators involve information loss, it might be advantageous to use multiple operators. A combination of orthogonal features could on the one hand capture more information and on the other increase the probability that one of them is discriminative for the specific modeling task.

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]

These 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 numeric aggregates, since they can better trade off instance discriminability and similarity.
- Use multiple aggregates to capture different notions of similarity.

We present in Section 3.3 a novel aggregation approach based on these principles, that is particularly appropriate for high-dimensional categorical variables.

3. Aggregation for relational learning

To provide context for the presentation of new aggregates, and the basis for a comprehensive empirical analysis of aggregation-based attribute construction, we will present briefly a learning system that can be applied to non-trivial relational domains. ACORA (Automated Construction of Relational Attributes) is a 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,

[3] Model induction methods suitable for high-dimensional input spaces may confer an advantage for such cases, as they often do for text. However, the transformation is not trivial. Since the input space is structured, it may be beneficial to tag values with the relation (chain) linking them to the target variable—author-smith versus cited-author-smith versus coauthor-smith, etc. Also, for relational problems even producing single-number aggregations can lead to a large number of features if there are moderately many relation chains to consider. Alternatively, such methods may be a useful alternative to this paper's methods for creating aggregation features: for example, include features representing the classification score based on authors alone, based on cited-author, based on coauthor, etc.

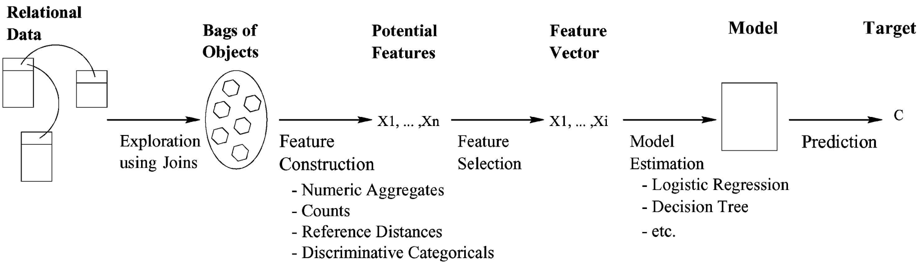

Fig. 2 ACORA's transformation process with four transformation steps: exploration, feature construction, feature selection, model estimation, and prediction. The first two (exploration and feature construction) transform the originally relational task (multiple tables with one-to-many relationships)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

- feature selection, and
- model estimation.

Figure 3 outlines the ACORA algorithm in pseudocode. Since the focus of this work is on aggregation we will concentrate on distribution-based aggregation assuming bags of values. Producing such bags of related objects requires the construction of a domain graph where the nodes represent the tables and the edges capture links between tables through identifiers; this is explained in more detail in Appendix A. 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 (Quinlan, 1993), and naive Bayes using WEKA (Witten & Frank, 1999), and are not discussed further in this paper.

The main idea behind the feature construction is to store meta-data on the (class-conditional) distributions of attributes' values, and then to use vector distances to compare the bags of values associated with particular cases to these distributional meta-data. In order to describe this precisely, we first must introduce some formal notation.

3.1. Setup and notation

A relational probability estimation (or classification) task is defined by a set of tables Q, R, ... (denoted by uppercase letters), including a particular **target table** T in a multi-relational database RDB. Every table R contains rows r (denoted in lowercase). The rows t of T are the **target entities** or **target cases**. Each table R has n_{R} fields and a row r represents the vector of field-values $r = (r.f_1, \ldots, r.f n_{\mathrm{R}})$ for a particular entity, which we will shorten to $r = (r.1, \ldots, r.n_{\mathrm{R}})$. Thus, $R.f$ denotes a field variable in table R, and $r.f$ denotes the value of $R.f$ for entity r.

The **domain** or **type**, $D(R.j)$, of field j in table R is either $\mathbb{R}$ in the case of numeric attributes, or the set of values that a categorical attribute $R.j$ can assume; in cases where this is not known a priori, we define $D(R.j) = \bigcup_{e \in R} r.j$, the set of values that are observed in field j across all rows r of table R. The **cardinality** $|D(R.j)|$ of a categorical attribute is equal to the number of distinct values that the attribute can take.

One particular attribute $T.c$ in the target table T is the class label for which a model is to be learned given all the information in RDB. We will consider binary classification where $D(T.c) = \{0, 1\}$. The main distinction between relational and propositional model induction

ACORA Algorithm

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

```
Read specification and build domain graph $\mathcal{G}$
Initialize breadth-first list $\mathcal{L}$ with target table: $\mathcal{L} = \{T\}$
Initialize feature table $F$ =non-identifier attributes($T$)
Loop
    $Q$ = First($\mathcal{L}$)
    Foreach table $R$ in $RDB$ related to $Q$ in $\mathcal{G}$ through some identifiers $(Q.l, R.k)$
        $J$ = Join $R$ and $Q$ under the condition $R.k$=$Q.l$
        Foreach attribute $R.j, j \neq k$                                (Section 3.3)
            Foreach target observation $t$
                Foreach applicable aggregation operator $\mathcal{A}$
                    Construct $\mathcal{A}(\mathcal{B}_{R.j}(t))$ where $\mathcal{B}_{R.j}(t)$ is the bag of values of attribute
                    $R.j$ related to target entity $t$.
                End Foreach
                Append aggregates $\mathcal{A}$ as attributes to $F$
            End Foreach
            Append $J$ to queue $\mathcal{L}$
        End Foreach
        if (stopping criterion) GOTO Select Features
    End Foreach
End Loop
Select Features $SF$ from $F$
Build propositional model from $SF$
```

Fig. 3 Pseudocode of the ACORA algorithm

is the additional information in tables of *RDB* other than *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 *R.k* to be unique across all rows in table *R* (the cardinality of the attribute is equal to the number of rows in the table). A link to information in another table *Q* is established if that key *R.k* also appears as *Q.l* in another table *Q*, 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 $ER(D(R.k), D(Q.l))$. 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** for 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 the identifier is unique. An example of an identifier that is not a key is the ISBN field in the transaction table in Figure 1.

Without semantic information about the particular domain it is impossible to say whether a particular field 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. $R.k$ is an **identifier** if $D(R.k) \neq \mathbb{R}$ and
$\exists\, Q.l$ with a cardinality $\geq I_{\text{MIN}}$ and $ER(D(R.k), D(Q.l))$

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 capture the same set of attributes. If I_{MIN} is at most 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 fields) 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.[4]

A **direct relationship** between entities is a pair of identifier fields $(Q.l, R.k)$ of equivalent type. For the modeling task we are mostly interested in entities that are related directly or indirectly to the cases in the target table T. Indirect relationships are captured by chains of identifier pairs such that the chain starts from the target table T and the second attribute of a pair is in the same table as the first attribute of the next pair: $(T.n, Q.m).(Q.l, R.k)\ldots$. The **bag** $\mathcal{B}$ of *objects* related to a case t in T under a relationship $(T.n, R.k)$ is defined as $\mathcal{B}_R(t) = \{r \mid t.n = r.k\}$ and the bag of related *values* of field $R.j$ is defined as $\mathcal{B}_{R.j}(t) = \{r.j \mid t.n = r.k\}$. For simplicity of notation we present this definition only for direct relationships and do not even index the bag by the the full details of the underlying relationship, but only by the final table; the extension to indirect relationships is straightforward. The reader should generally keep in mind that $\mathcal{B}$ is not defined globally but for a specific relationship chain.

3.2. Simple aggregation

In order to apply traditional induction techniques, 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 example is the aggregation of the ISBN attribute, which we assume has cardinality larger than I_{MIN}. An aggregation operator $\mathcal{A}$ provides a mapping from bags of values $\mathcal{B}_{R.j}(t)$ to $\mathbb{R}$, to $\mathbb{N}$, or to the original type of the field $D(R.j)$. Simple aggregation operators for bags of categorical attributes are the *COUNT*, value counts for all possible values $v \in D(R.j)$, and the *MODE*. The $COUNT = |\mathcal{B}_{R.j}(t)|$ captures only the size of the bag. $COUNT_v$ for a particular value v is the number of times value v appeared in the bag $\mathcal{B}_{R.j}(t)$, and the *MODE* is the value v that appears most often in $\mathcal{B}_{R.j}(t)$. In the example, $MODE(\mathcal{B}_{\text{Transaction.TYPE}}(C2, 1))$ = 'Non-Fiction' for the bag of values from the TYPE field in the Transaction table, related to the case 'C2,1' in the customer table through the CID identifier.

None of these simple aggregates is appropriate for high-cardinality fields. For example, since most customers buy a book only once, for bags of ISBNs there will be no well-defined

[4] 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 relationships will be considered.

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 5), but rarely addressed explicitly. Some relational learning systems (Krogel & Wrobel, 2001) only consider attributes with cardinality of less than n, typically below 50; Woznica et al. (2004) define **standard attributes** excluding keys, and many ILP (Muggleton & DeRaedt, 1994) systems require the explicit identification of the categorical values that may be considered for equality tests, leaving the selection to the user.

3.3. Aggregation using distributional meta-data

Aggregation summarizes a set or a distribution of values. As we will describe in detail, 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.

Although its use is not as widespread as in statistical hypothesis testing, distributional meta-data 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 (Fawcett & Provost, 1997). 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.

Estimating a distribution from each 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.

Table 1 presents the result of the join (on CID) of the two tables in our example database (step 7 in the pseudocode). Consider the bag $\mathcal{B}_{\text{Transaction}}(C2, 1)$ of related transactions for customer C2:

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

The objective of an aggregation operator $\mathcal{A}$ 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 $\mathcal{B}_{\text{TYPE}}(C2, 1) = \langle\text{Non-Fiction,Non-Fiction,Fiction}\rangle$, $\mathcal{B}_{\textbf{ISBN}}(C2, 1) =$

[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).

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 entity (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

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-iction	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

$\langle 231{,}523{,}856\rangle$, and $\mathcal{B}_{\text{PRICE}}(C2, 1) = \langle 12.99{,}9.49{,}4.99\rangle$. Aggregating each bag of attributes separately brings into play an assumption of class-conditional independence between attributes of related entities (Perlich & Provost, 2003). ACORA may apply one or more aggregation operators to each bag. Simple operators that are applicable to bags of numeric attributes such as $\mathcal{B}_{\text{Transactions.PRICE}}$, or $\mathcal{B}_{\text{PRICE}}$ for short, include the $SUM = \sum c \in \mathcal{B}_{\text{PRICE}}$ or the $MEAN = SUM/|\mathcal{B}_{\text{PRICE}}|$. Consider on the other hand $B_{\text{ISBN}}(C2, 1) = \langle 231{,}523{,}856\rangle$. ISBN is an example of a bag of values of an attribute with high cardinality, where the *MODE* is not meaningful because the bag does not contain a "most common" element. The high cardinality also prevents the construction of counts for each value, because counts for each possible ISBN would result in a very sparse feature vector with a length equal to the cardinality of the attribute (often much larger than the number of training examples), which would be unsuitable for model induction.

3.3.1. Reference vectors and distributions

The motivation for the new aggregation operators presented in the sequel is twofold: (1) to deal with bags of high-cardinality categorical attributes for which no satisfactory aggregation operators are available, and (2) to develop aggregation operators that satisfy the principles outlined in Section 2 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 distribution (or reference bag)**. Reference bags/distributions are constructed as follows. Let us define a **case vector** $CV_{R.j}(t)$ as the vector representation of a bag of categorical values $B_{R.j}(t)$ related to target case t. Specifically, given an ordering, $N : D(R.j) \rightarrow \mathbb{N}$, and a particular value v of field $R.j$, the value of $CV_{R.j}(t)$ at position $N(v\ )$ is equal to the number of occurrences of value v in the bag.

$$CV_{R.j}(t)[N(v)] = COUNT_v \tag{1}$$

For example $CV_{\text{Transaction TYPE}}(C2{,}1) = [2{,}1]$ for $\mathcal{B}_{\text{Transaction.TYPE}}(C2, 1) = \langle$Non-Fiction,Non-Fiction,Fiction$\rangle$, under the order N(Non-Fiction) = 1, N(Fiction) = 2.

Based on the case vectors in the training data, the algorithm constructs two class-conditional **reference vectors** RV^0 and RV^1 and an unconditional reference vector RV^*:

Table 2 "Case vector" representation of the bags of the TYPE and ISBN attributes for each target case ($C1$ to $C4$) after the exploration in Table 1. The vector components denote the counts of how often a value appeared in entities related to the target case

TYPE	Non-fiction	Fiction
$CV(C1, 0)$	0	1
$CV(C2, 1)$	2	1
$CV(C3, 1)$	1	0
$CV(C4, 0)$	1	3

ISBN	231	475	523	673	856	937
$CV(C1, 0)$	0	0	1	0	0	0
$CV(C2, 1)$	1	0	1	0	1	0
$CV(C3, 1)$	1	0	0	0	0	0
$CV(C4, 0)$	0	1	0	1	1	1

$$RV^{0}_{R.j}[N(v)] = \frac{1}{s_0} \sum_{\{t|t.c=0\}} CV_{R.j}(t)[N(v)] \tag{2}$$

$$RV^{1}_{R.j}[N(v)] = \frac{1}{s_1} \sum_{\{t|t.c=1\}} CV_{R.j}(t)[N(v)] \tag{3}$$

$$RV^{*}_{R.j}[N(v)] = \frac{1}{s_1 + s_0} \sum_{t} CV_{R.j}(t)[N(v)] \tag{4}$$

where s_0 is the number of negative target cases and s_1 is the number of positive target cases, and $[k]$ denotes the kth component of the vector. $RV_{\mathrm{Rj}}^{1}[N(v)]$ is the average number of occurrences of value v related to a positive target case ($t.c = 1$) and $RV_{R.j}{}^{0}[N(v)]$ the average number of occurrences of a value v related to a negative target case ($t.c = 0$). $RV_{R.j}$ $[N(v)]$ is the average number of occurrences of the value related to any target case. We also compute **distribution vectors** DV^{0}, DV^{1} and $DV*$ that approximate the class-conditional and unconditional distributions from which the data would have been drawn:

$$DV^{0}_{R.j}[N(v)] = \frac{1}{\sum_{\{t|t.c=0\}} b_t} \sum_{\{t|t.c=0\}} CV_{R.j}(t)[N(v)] \tag{5}$$

$$DV^{1}_{R.j}[N(v)] = \frac{1}{\sum_{\{t|t.c=1\}} b_t} \sum_{\{t|t.c=1\}} CV_{R.j}(t)[N(v)] \tag{6}$$

$$DV^{*}_{R.j}[N(v)] = \frac{1}{\sum_{\{t\in T} b_t\}} \sum_{\{t\in T\}} CV_{R.j}(t)[N(v)] \tag{7}$$

where b_t is the number of values related to target case t (the size of bag $\mathcal{B}_{R.j}(t)$). For the example, the case vectors for TYPE and ISBN are shown in Table 2 and the reference vectors and distributions in Table 3. Extend the pseudocode of step 8:

Table 3 Reference vectors and reference distributions for the TYPE and ISBN attributes for objects in Table 1:class-conditional positive DV^1, class-conditional negative DV^0, and unconditional distribution DV^*. The reference vectors and reference distributions capture the same information, but with different normalizations: division by the number of target cases or by the number of related entities

TYPE	Non-fiction	Fiction
RV^1	1.5	0.5
RV^0	0.5	2.0
RV^*	1.0	1.25
DV^1	0.75	0.25
DV^0	0.20	0.80
DV^*	0.44	0.55

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

```
Input: All bags $\mathcal{B}_{R.j}(t)$ for attribute $j \neq k$ of all target cases $t$.
8.1    Foreach target case $t$ estimate $CV_{R.j}(t)$
8.2    Estimate $RV^0_{R.j}, RV^1_{R.j}, RV^*_{R.j}, DV^0_{R.j}, DV^1_{R.j}, DV^*_{R.j}$
```

3.3.2. Distances to reference vectors and distributions

The aggregation in step 11 of ACORA's pseudocode 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 categorical values of attribute $R.j$ as:

$$A(\mathcal{B}_{R.j}(t)) = DIST(RV, CV_{R.j}(t)) \tag{8}$$

where $DIST$ can be any vector distance and $RV \in \{RV^0_{R.j}, RV^1_{R.j}, RV^*_{R.j}, DV^0_{R.j}, DV^1_{R.j}, DV^*_{R.j}\}$. 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. In the case of cosine distance the normalization (RV^0 vs. DV^0) is irrelevant, since cosine normalizes by the vector length.

Consider the result of step 12 of the algorithm on our example (Table 4), where two new attributes are appended to the original feature vector in the target table, using cosine distance to RV^1 for the bags of the TYPE and the ISBN attributes. Both features appear highly predictive (of course the predictive power has to be evaluated in terms of out-of-sample performance for test cases that were not used to construct RV^0 and RV^1).

Observe the properties of these operators in light of the principles derived in Section 2: (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,

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

t.CID	t.CLASS	Cosine($RV^1{}_{\mathrm{TYPE}}$, $CV_{\mathrm{TYPE}}(t)$)	$Cosine$($RV^1{}_{\mathrm{ISBN}}$, $CV_{\mathrm{ISBN}}(t)$)
$C1$	0	0.316	0.408
$C2$	1	0.989	0.942
$C3$	1	0.948	0.816
$C4$	0	0.601	0.204

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, because 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.

3.3.3. Simpler but related aggregate features

An alternative solution to deal with bags of values from high-cardinality attributes is to select a smaller subset of values for which the counts are used as new features. This poses the question of a suitable criterion for selection, and the distributional meta-data can be brought to bear. For example, a simple selection criterion is high overall frequency of a value. ACORA constructs in addition to the vector-distance features, the top n values v for which $DV*(N(v))$ was largest.

However, the principles in Section 2 suggest choosing the *most discriminative* values based on the target prediction task. Specifically, ACORA uses the class-conditional reference vectors RV^0 and RV^1 (or the distributions DV^0 and DV^1) to select those that show the largest absolute values for $RV^1 - RV^0$. For example, the most discriminative TYPE value in the example is ‘Fiction’ with a difference of 1.5 in Table 3.

For numeric attributes, ACORA provides straightforward aggregates: *MIN*, *MAX*, *SUM*, *MEAN*, and *VARIANCE* It also discretizes numeric attributes (equal-frequency binning) and estimates class-conditional distributions and distances, similar to the procedure for categorical attributes described in Section 3.3.1. 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.

3.4. **A Bayesian justification: A relational fixed-effect model**

We suggested distance-based aggregates to address a particular problem: the aggregation of categorical variables of high cardinality. The empirical results in Section 4 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. Aside from empirical evidence of superior modeling performance, we now show that the distance-based aggregation operators can be derived as components of a “relational fixed-effect model” with a Bayesian foundation.

Statistical estimation contrasts *random-effect* models with *fixed-effect* models (DerSimonian & Laird, 1986). In a random-effect model, the model parameters are not assumed to be constant but instead to be drawn randomly from a distribution for different observations. An analogy can be drawn to the difference between our aggregates and the traditional simple aggregates from Section 3.2. Simple aggregates estimate parameters from a distribution for each bag. This is similar to a random effect model. Our aggregates on the other hand can be seen as a relational *fixed-effect* model: we assume the existence of only two fixed distributions, one for each of the two classes.[6] Under this assumption the number of parameters decreases by a factor of $n/2$ where n is the number of training examples. More specifically, we define a **relational fixed-effect model** with the assumption that all bags of objects related to positive target cases are sampled from one distribution DV^1 and all objects related to negative target cases are drawn from another distribution DV^0. Thus it may become possible to compute reliable estimates of reference distributions DV^1 and DV^0, even in the case of categorical attributes of high cardinality, by combining all bags related to positive/negative cases to estimate DV^1/DV^0.

In a relational context, a target object t is described not only by its own attributes. It also has an identifier (CID in our example) that maps into bags of related objects from different background tables. Given a target object t with a feature vector[7] and a set of bags of related objects from different relationships $(t.1, \ldots, t.n_t, \mathcal{B}_R(t), \ldots, \mathcal{B}_Q(t))$, via Bayes' rule one can express the probability of class c as

$$P(c\,|t) = P(c\,|t.1, \ldots, t.n_t, \mathcal{B}_R(t), \ldots, \mathcal{B}_Q(t)) \tag{9}$$

$$= P(t.1, \ldots, t.n_t, \mathcal{B}_R(t), \ldots, \mathcal{B}_Q(t)|c) * P(c)/P(t). \tag{10}$$

Making the familiar assumption of class-conditional independence of the attributes $t1, \ldots$ $,tn_{\mathrm{t}}$ *and* of all bags $\mathcal{B}_*$ of related objects allows rewriting the above expression as

$$P(c|t) = \prod_i P(t.i \mid c) * \prod_{\mathcal{B}} P(\mathcal{B}_*(t) \mid c) * P(c)/P(t). \tag{11}$$

Assuming that the elements r of each particular bag of related objects $\mathcal{B}_R(t)$ are drawn independently, we can rewrite the probability of observing that bag as

$$P(\mathcal{B}_R \mid c) = \prod_{r \in \mathcal{B}_R(t)} P(r \mid c). \tag{12}$$

Assuming again class-conditional independence of all attributes $r1, \ldots\ldots\ldots, rn_{\mathrm{p}}$ of all *related* entities r, we can finally estimate the class-conditional probability of a bag of values from the training data as

$$P(\mathcal{B}_R(t) \mid c) = \prod_{r \in \mathcal{B}_R(t)} * \prod_j P(r.j \mid c). \tag{13}$$

[6] More than two distributions would be used for multiclass problems, or could be generated via domain knowledge or clustering.

[7] Excluding the class label and the identifier.

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

$$P(\mathcal{B}_R(t) \mid c) = \prod_j * \prod_{r.j \in \mathcal{B}_{R.j}(t)} P(r.j \mid c). \tag{14}$$

This second part of the product has a clear probabilistic interpretation: it is the non-normalized (ignoring $P(c)$ and $P(t)$) probability of observing the bag of values $\mathcal{B}_{R.j}(t)$ of attribute $R.j$ given the class c This non-normalized conditional probability or *likelihood* can be seen as a particular choice of vector distance[8] and can be estimated in the previous notation as:

$$LH(DV^c, CV) = P(\mathcal{B}_{R.j}(t) \mid c) = \prod_{r.j \in \mathcal{B}_{R.j}(t)} P(r.j|c) = \prod_i DV^c[i]^{CV[i]} \tag{15}$$

where i ranges over the set of possible values for the bagged attribute.

Thus, for the particular choice of likelihood (*LH*) as the distance function, ACORA's aggregation approach can be given a theoretical foundation within a general relational Bayesian framework similar to that of Flach and Lachiche (2004).

This derivation not only 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 will come typically at a price: modeling will become increasingly prone to overfitting because the search space expands rapidly. This calls for strong constraints on the search space, as typically are provided for ILP systems in the declarative language bias. We discussed this tradeoff previously (Perlich & Provost, 2003) in the context of noisy domains.

3.5. Implications for learning from identifier attributes

We show in our empirical results in Section 4 the importance of including aggregates of identifiers. The following discussion is an analysis of the special properties of identifiers and why aggregates of identifiers and in particular additive distances like cosine can achieve such performance improvements.

Identifiers are categorical attributes with high cardinality. In our example problem we have two such attributes: CID, the identifier of customers, and ISBN, the identifier of books. The task of classifying customers based on the target table 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 predictive out-of-sample. For example, buying Jacques Pépin's latest book may increase the estimated likelihood that the customer would join a

[8] Actually a similarity.

cookbook club; in a different domain, calling from a particular cell site (location) may greatly increase the likelihood of fraud. Such identifiers may be 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 $R.j$, such as ISBN, allows the estimation of class-conditional probabilities $P(r.j|c)$.

Beyond the immediate relevance of particular identities (e.g., Pépin's book), identifier attributes have a special property: they represent implicitly all characteristics of an object. Indeed, the *identity* of a related object (a particular cell site) 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 Eq. (12) and even permits learning from unobserved characteristics.

An object identifier Rj like ISBN stands for all characteristics of the an object. If observed, these characteristics would appear in another table S as attributes ($S.1, \ldots, S.n_{\mathrm{s}}$). Technically, there exists a functional mapping[9] F that maps the identifier to a set of values: $F(r.j) \rightarrow (s.1, \ldots, s.n_{\mathrm{s}})$. We can express the joint class-conditional probability (*without* the independence assumption) of a particular object feature-vector without the identifier field as the sum of the class-conditional probabilities of all objects (represented by their identifiers $r.j$) with the same feature vector:

$$P(s.1, \ldots, s.n_s \mid c) = \sum_{\{r.j|F(r.j)=(s.1,\ldots,s.n_r)\}} P(r.j \mid c) \tag{16}$$

If F is an isomorphism (i.e., no two objects have the same feature vector) the sum vanishes and $P(s.1, \ldots, s.n_s \mid c) = P(r.j \mid c)$. Estimating $P(r.j \mid c)$ therefore provides information about the joint probability of all its attributes ($s1, \ldots, s\ n_{\mathrm{s}}$).

A similar argument can be made for an unobserved attribute $s.u$ (e.g., association with an organization engaging in fraud). In fact, it may be the case that no attribute of the object s was observed and no table S was recorded, as is the case for ISBN in our example. There is nevertheless the dependency $F'(r.j) \rightarrow su$, for some function F', and the relevant class-conditional probability is equal to the sum over all identifiers with the same (unobserved) value:

$$P(s.u \mid c) = \sum_{\{r.j|F'(r.j)=s.u\}} P(r.j \mid c). \tag{17}$$

Given that $s.u$ is not observable, it is impossible to decide which elements belong into the sum. If however $s.u$ is a perfect predictor—i.e., every value of su appears only for objects related to target cases of one class c—the class-conditional probability $P(r.j \mid c)$ will be non-zero for only one class c In that case the restricted sum in Equ (17) is equal to the total sum over the class-conditional probabilities of all identifier values:

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

$$\sum_{\{r|F'(r.j)=s.u\}} P(r.j \mid c) = \sum_{r} P(r.j \mid c). \tag{18}$$

Note that the total sum over the class-conditional probabilities of all related identifier values now equals the cosine distance between DV^{c} and a special case vector CV^{su} that corresponds to a bag containing all identifiers with value *su* prior to normalization[10] by vector length, because $DV^{c}[N(r.j)]$ is an estimate of $P(r.j \mid c)$ and $CV[N(r.j)]$ is typically 1 or 0 for identifier attributes such as ISBN. The cosine distance for a particular bag $CV(t)$ is a biased[11] estimate of $P(s.u|c)$ since the bag will typically only consist of a subset of all identifiers with value *s.u*

$$\text{cosine}(DV^{c}_{R.j}, CV) = \frac{1}{||CV||} \sum_{i} DV^{c}_{R.j}[i] * CV_{R.j}[i] \tag{19}$$

So far we have assumed a perfect predictor attribute *S.u*. The overlap between the two class-conditional distributions DV^{0} and DV^{1} of the identifier is a measure of the predictive power of *Su* and also how strongly the total sum in the cosine distance deviates from the correct restricted sum in Eq. (17). 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 4.

Although this view is promising, issues remain. It maybe hard to estimate $P(r.j|c)$ 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 need 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 DV^{0} and DV^{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 4.

Identifiers have other interesting properties. Identities may often be the *cause* of relational autocorrelation (Jensen & Neville, 2002). 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 autocorrelation between customers who 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 may introduce a “natural” Markov barrier that reduces or eliminates the need to extend the search for related entities further than to the direct neighbors.[12] We present some evidence of this phenomenon in Section 4.3.3.

[10] The effect of normalization can be neglected, since the length of DV_{c} is 1 and the length of CV is the same for both the class-conditional positive and class-conditional negative cosine distances.

[11] We underestimate $P(s.u|c)$ 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.

[12] In cases with strong class-label autocorrelation, such a barrier often can be provided by class labels of related instances (Jensen et al., 2004; Macskassy & Provost, 2004).

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 attribute and $D(y) = \mathbb{R}$ identifies numeric attributes

Domain	Table: Size	Attribute type description	Size
XOR	T: 10000	C(tid) = 10000, C(c) = 2	Train: 8000
	O: 55000	C(oid) = 1000, C(tid) = 10000	Test: 2000
AND	T: 10000	C(tid) = 10000, C(c) = 2	Train: 8000
	O: 55000	C(oid) = 1000, 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

4. Empirical results

After describing the experimental setup, Section 4.3 presents 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 4.4).

4.1. Domains

Our experiments are based on eight relational domains that are described in more detail in Appendix B. They typical are 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 (Fawcett & Provost, 1997; Cortes et al., 2002). 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.

4.2. Methodology

Our main objective is to demonstrate that distribution-based vector distances for aggregation generalize when simple aggregates like *MODE* or $COUNT_v$ for all values are inapplicable or inadequate. In order to provide a solid baseline we extend these simple aggregates for use in the presence of attributes with high cardinality: ACORA constructs $COUNT_{\mathrm{v}}$ for the 10 most common values (an extended *MODE* based on the meta-data) and counts for all values if the number of distinct values is at most 50, as suggested by Krogel and Wrobel (2003). ACORA generally includes an attribute for each bag 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 (Witten & Frank, 1999) to estimate probabilities of class membership from all features. Using decision trees (including the differences of distances as suggested in Section 4.3.1) 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 runs the number of constructed features remains relatively small.

Evaluation. The generalization performance is evaluated in terms of the AUC: area under the ROC curve (Bradley, 1997). All results represent out-of-sample generalization performance on test sets averaged over 10 runs. The objects in the target table for each run are 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.

4.3. 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 with high-dimensional categorical variables was motivated by three arguments that are considered in the sequel:

- Target-dependent aggregates, such as vector distances to class-conditional reference vectors, exhibit task-specific similarity;
- This task-specific similarity, in combination with the instance discriminability conferred by using numeric distances, improves generalization performance;
- Aggregating based on vector distances allows learning from identifier attributes, which hold certain special properties (viz., proxying for: unseen features, interdependent features, and information farther away in the network). Coalescing information from many identifiers can improve over including only particular identifiers.

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 denotes a target case

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 3.3.3) 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(DV^1, $CV(t)$), Cosine(DV^0, $CV(t)$)
Mahalanobis	Mahalanobis(RV^1, $CV(t)$), Mahalanobis(RV^0, $CV(t)$)
Euclidean	Euclidean(RV^1, $CV(t)$), Euclidean(RV^0, $CV(t)$)
Likelihood	Likelihood(DV^1, $CV(t)$), Likelihood(DV^0, $CV(t)$)
UCVD	All unconditional vector distances: Cosine(DVˆ*, $CV(t)$), Mahalanobis(DVˆ*, $CV(t)$), Euclidean(DVˆ*, $CV(t)$), Likelihood(DV, $CV(t)$)
CCVD	All class-conditional vector distances: Cosine(DV^1, $CV(t)$), Cosine(DV^0, $CV(t)$), Euclidean(DV^1, $CV(t)$), Euclidean(DV^0, $CV(t)$), Mahalanobis(DV^1, $CV(t)$), Mahalanobis(DV^0, $CV(t)$), Likelihood(DV^1, $CV(t)$), Likelihood(DV^0, $CV(t)$)
DCCVD	All differences of class-conditional vector distances: Cosine(DV^1, $CV(t)$) – Cosine(DV^0, $CV(t)$), Mahalanobis(DV^1, $CV(t)$) – Mahalanobis(DV^0, $CV(t)$), Euclidean(DV^1, $CV(t)$) – Euclidean(DV^0, $CV(t)$), Likelihood(DV^1, $CV(t)$)– Likelihood(DV^0, $CV(t)$)

We also argued that using multiple aggregates can improve generalization performance. As we will see, this point is not supported as strongly by the experimental results.

4.3.1. Task-specific similarity

We argued in Section 2 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). For the XOR problem, Figure 4 shows on the left 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 has a different bag of identifiers, when 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 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 (Perlich, 2005b). Figure 4 shows on the right the distribution of the differences for cases of both classes with an optimal splitting point around zero.

Figure 5 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.

4.3.2. Comparative generalization performance

We now show that the use of aggregations based on (distances to) distributional meta-data adds generalization power over traditional aggregations (and over 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 are inapplicable in most of the domains. (Entries with a * denote cases where the 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 distances, class-conditional dis-

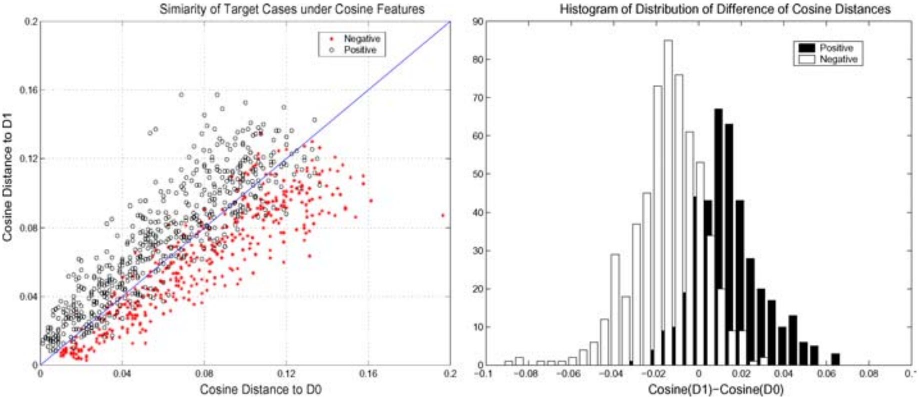

Fig. 4 In the left plot, the two-dimensional feature space (XOR domain) of the class-conditional cosine distances for the identifiers of related entities shows (i) high instance-discriminability (different target cases are assigned unique points in this space) and (ii) task-specific similarity, where 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

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

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)

tances, and most-discriminative counts). For all domains the aggregation of high-dimensional categorical attributes using distances to 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. This 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.

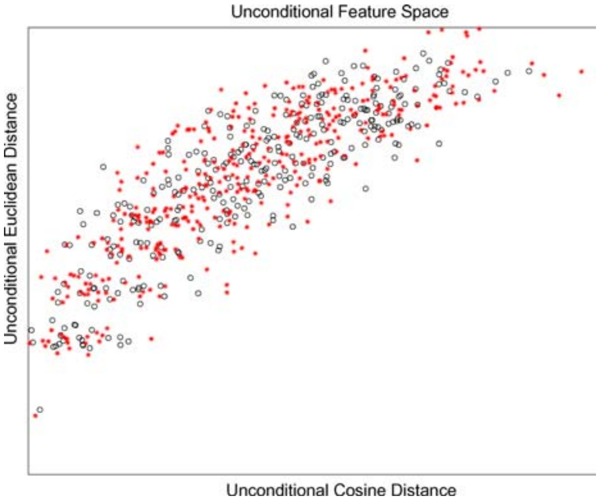

Fig. 5 The two-dimensional feature space of the unconditional cosine and Euclidean distances still shows high instance-discriminability, but lacks task-specific similarity. Positive cases are as similar to negative cases as they are to other positive cases. As a result these features have no discriminative power

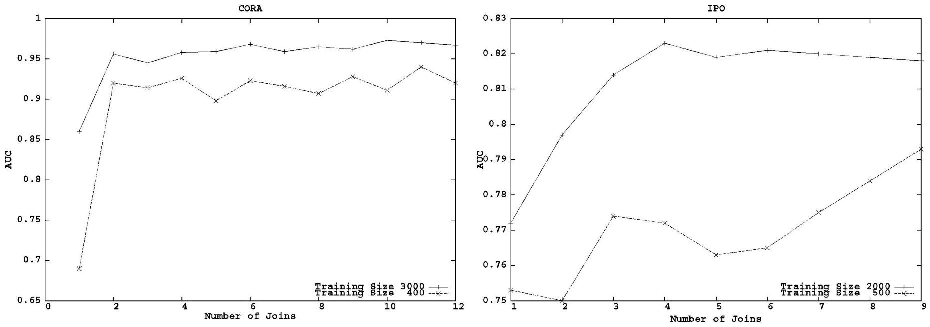

Fig. 6 Ranking performance (AUC) on the CORA (left) and IPO (right) domains as a function of the number of joins for two different training sizes (400 and 3000). Beyond depth = 1 (see Figure 12 in Appendix A) no new discriminative information is found for CORA, because the depth-one identifier attributes 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

For the artificial domains and the synthetic fraud domain, neither type of distribution-based count (MCC nor MDC) provides any predictive power. This will be explained in Section 4.4.2. 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.

Finally, although we conjectured when discussing the design criteria (Section 2) that using multiple distance measures would be valuable, we find only weak support. Combining MDC and CCVD improves slightly. CCVD itself combines various distance measures; comparing it with the performance of its component measures alone (not shown here) shows that for the most part the combination only improves slightly over the overall best single distance measure (cosine). However, CCVD is more robust across domains than any individual distance measure, yielding the highest accuracy on 6 of 8 domains, and only underperforming the individual best[13] by a percentage point or two on the other two domains. Contrast this with likelihood, which underperformed by a large margin on every domain.

In summary, 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.

4.3.3. 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

[13] Cosine on COOC; Euclidean on Fraud.

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 3.5 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 (Fawcett & Provost, 1997). 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.

Figure 6 shows 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, which we see for IPO. For the larger training size, the IPO curve flattens out after 4 joins, supporting the claim that (with enough training data) the identifiers proxy for deeper-linked information. Even for the smaller training size, CORA needs no more than two joins. Traversing two joins obtains information about a paper's authors and its particular citations (cf., the search graph in Appendix A, but not about the classes of those citations (for which a join back to Paper would be necessary). Just knowing the author(s) and the *particular* papers cited apparently is enough—which can be compared to the success that has been achieved by drawing inferences based on the classes of linked papers (Taskar et al., 2001; Macskassy & Provost, 2004).

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

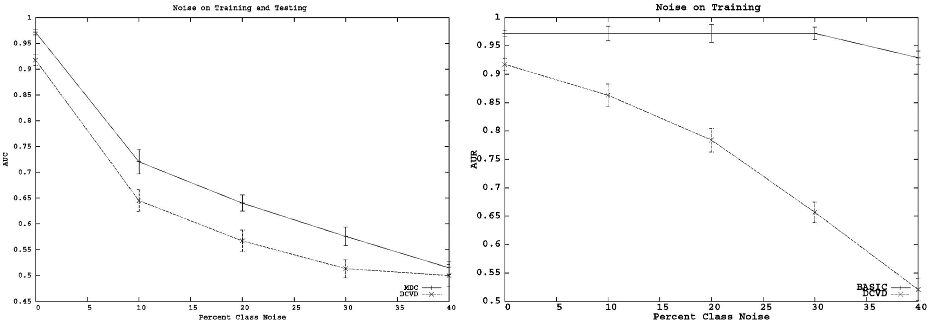

Fig. 7 Performance degradation for the AND domain *with x and y observed*, as a function of the amount of 0/1 class noise. In the left plot both training and test sets were corrupted; the right plot shows results using a noisy training set and clean test set, as a measure of the ability to recover the true concept

4.4.1. 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, if we allow x and y to be observed (in contrast with the main results). Recall from the discussion of the identifier attributes above that 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 class noise increases. The left plot in Figure 7 compares the sensitivity of CCVD and COUNTS to class-noise as a function of the noise level (p percent of training and test class labels are reassigned randomly from a uniform 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 7 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. 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. If x and y are observed, the COUNTS aggregation can combine information about x and y from all bags and therefore is not very sensitive to the random variation. 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.

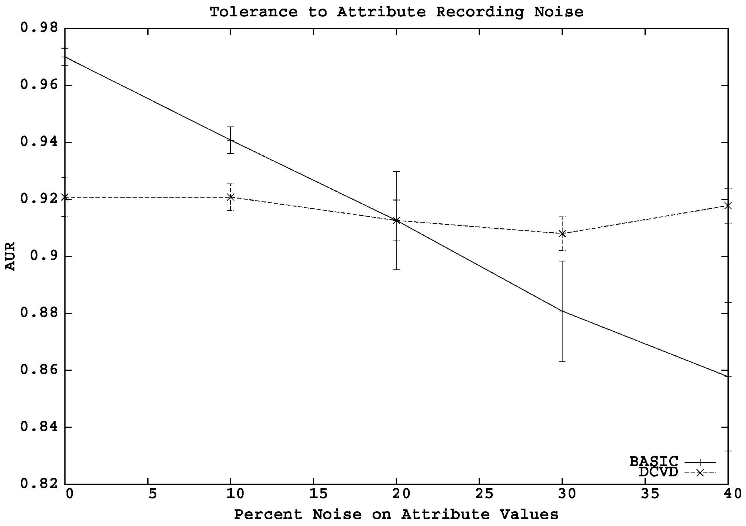

Fig. 8 Performance sensitivity on the AND domain to attribute recording noise for related entities. Since CCVD does not use the values of x and y (unobserved properties) it shows no performance decrease

In contrast, if *attribute* noise (misrecorded values of x and y) is present, we would expect the aggregates of identifier attributes to fare much better. Indeed, Figure 8 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 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.

4.4.2. Relational structure

Another potential point of sensitivity of the distribution-based aggregation methods is the structure of the 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,[14] and also that there is preferential attachment to particular entities (e.g., hubs on the Web).

To investigate s4nsitivity 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

[14] Jensen et al. (2003). discuss degree disparity and potential problems that it can cause for relational modeling.

Table 8 Measures of the skewedness (differences in the likelihood of a entity to be related to some target case) of the relational structure: counts of the links to the 5 most commonly linked entities, normalized by the number of target cases, and in the last column the number of links to the least common entity. 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.

Domain	1st	2nd	3rd	4th	5th	Least Common
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	0.1258	0.0945	0.0773	5

Table 9 Ranking performance (AUC) on the XOR domain for uniform distribution (XOR 1) and highly skewed distribution (XOR 4), including standard deviations across 10 experiments

Domain	COUNTS	MCC	UCVD	CCVD	MDC	MDC&CCVD
XOR 1	0.53 (0.018)	0.51 (0.02)	0.62 (0.02)	0.92 (0.008)	0.51 (0.004)	0.92 (0.008)
XOR 4	0.54 (0.02)	0.49 (0.04)	0.71 (0.012)	0.78 (0.007)	0.75 (0.011)	0.86 (0.007)

case can differ significantly across entities. Unfortunately this cannot be quantified easily as in the numeric case of a third moment. Table 8 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 9 compares the performances of the various aggregations on XOR 1 and XOR 4. 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.

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 9 shows the effects of the skew (see Table 8) in combination with the training size.

Observe the interesting pattern: for stronger skew, we see 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 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

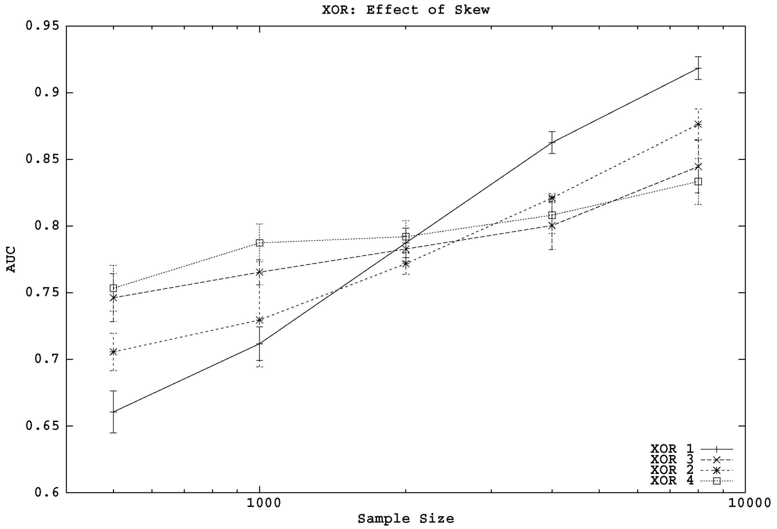

Fig. 9 Interaction effect of skew of relationship distribution and training size on ranking performance for the XOR domain. A stronger skew provides more useful information early, but the marginal value of additional training examples is lower

more information because the marginal information for each additional training case is larger (cf., the "Least Common" column in Table 8). The relatively low performance (compared with the uniform case) for XOR4 of CCVD in Table 9 is a result of the large training size in combination with a high skew.

4.4.3. Domain characteristics and performance with few training data

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).

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 (Brazdil et al., 1994; Perlich et al., 2003). 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 strong skew can improve performance with small training sets. The size of the bags determines the number of effective observations for the estimation of $P(t.id \mid c)$. 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 one percent of the target cases are positive, very few observations are available for $P(tid|1)$ and many for $P(t.id \mid 0)$; such class skew can be problematic if the minority class is much better

Table 10 Performance (AUC) using cosine distance with small training sets (250 examples) and interaction with skew (1st and Least Common, and where the latter equals 1 the number of values that appeared only once), unconditional prior of class 1, and average bag size

Domain	1st	Least common	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

defined ("customers who ...") than the majority class ("everyone else"), as is often the case. Table 10 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 show the skew. 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 $P(t.id \mid c)$ 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 4.4.2 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 analyzes 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).

4.5. 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., Koller & Pfeffer, 1998; Neville et al., 2003a; Popescul & Ungar, 2003). 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

Table 11 Accuracy comparison with logic-based relational classifiers (FOIL, Tilde, Lime, Progol), target features (TF), and using no relational information (Prop) as a function of training size on the IPO domain

Size	Prop	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

propositional model from the new features. There are, of course, exceptions. For example, RELAGGS (Krogel & Wrobel, 2001) would be outside of ACORA's expressive power since it combines Boolean conditions and aggregation and can form more complex aggregations (cf., hierarchy of aggregation complexity described by (Perlich & Provost, 2003).

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 automatically, and without those attributes only the few attributes in EBook and IPO and the 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 too would 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 including FOIL (Quinlan & Cameron-Jones, 1993), TILDE (Blockeel & Raedt, 1998), Lime (McCreath, 1999), and Progol (Muggleton, 2001). Since ILP systems typically (with the exception of TILDE) only predict the class, not the probability of class membership, we compare in Table 11 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 (Prop). ACORA uses a stopping criteria of depth = 3. 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 bank identifiers were not included as model constants.

The results in Table 11 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 (Prop), 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 bank identifiers 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 12) its AUC to our results from above.[15] 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

[15] On the IPO domain TILDE improved also in terms of accuracy over the performance without banks in Table 11 from 0.65 to 0.753.

Table 12 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

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)

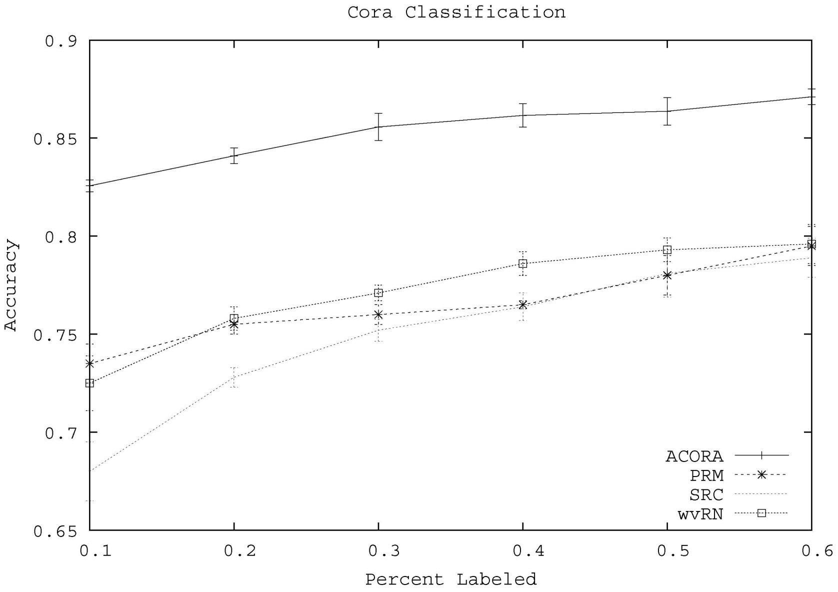

Fig. 10 Comparison of classification accuracy of ACORA using class-conditional distributions against a Probabilistic Relational Model (PRM) and a Simple Relational Classifier (SRC) on the CORA domain as a function of training size

provided identifier attributes. Note that 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 coalesce information across multiple identifier values is necessary to learn predictive models.

Figure 10 shows that using identifier attributes would likely have improved other published statistical relational learning approaches as well. For the Cora domain, the figure shows classification accuracies as a function of training size. ACORA estimates seven separate binary classification models using class-conditional distributions for each of the seven classes and predicts the final class with the highest probability score across the seven model predictions. The figure compares ACORA to prior published results by (Taskar et al., 2001) using Probabilistic Relational Models (PRM, Koller & Pfeffer, 1998), based on both text and relational information, a Simple Relational Classifier (SRC) by Macskassy and Provost (2003) 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, and wvRN (Macskassy & Provost, 2004), a later version of SRC. 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.

A further indicator of ACORA's strong performance on difficult and noisy domains in comparison to other relational learning approaches was its winning entry at the 2005 ILP Challenge (Perlich, 2005a). The task was the classification of yeast genes into various functional classes. The model analysis showed that the functional class of very similar genes along with the identity of similar proteins were the strongest indicators of protein function.

5. 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.

Aggregation has been identified as a fundamental problem for relational learning from real-world data (Goldberg & Senator, 1995), however machine learning research has considered only a limited set of aggregation operators. Statistical relational learning often treats aggregation as a preprocessing step that is independent of the model estimation process. Inductive Logic Programming (Muggleton & DeRaedt, 1994) also typically considers only a special sort of aggregation for one-to-many relationships—existential quantification—which is an integral part of the search through the model space.

Propositionalization (e.g., Knobbe et al., 2001; Krogel & Wrobel, 2001; Krogel & Wrobel, 2003) has long recognized the essential role of aggregation in relational modeling, focusing specifically on the effect of aggregation choices and parameters, and yielding promising empirical results on noisy real-world domains. The numeric aggregates used by Knobbe et al., (2001) outperform the ILP systems FOIL (Quinlan & Cameron-Jones, 1993), Tilde (Blockeel & Raedt, 1998), and Progol (Muggleton, 2001) on a noisy financial task (PKDD-CUP 2000). Krogel and Wrobel (2001, 2003) show similar results on the financial task and a customer-classification problem (ECML 1998 discovery challenge) in comparison to Progol and Dinus (Lavrač & Ďzeroski,1994), a logic-based propositionalization approach. Similar work by Krogel et al. (2003) 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. Previous results (Perlich & Provost, 2003) indicate 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 (Kirsten et al., 2000) 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 the paper by Gärtner (2003). Structured kernels estimate distances between complex objects and typically are tailored towards a particular domain. This distance estimation also involves aggregation and often uses sums.

Statistical relational learning approaches (Neville et al., 2003c; Jensen & Getoor, 2003) include network models as well as upgrades of propositional models (e.g., Probabilistic Relational Models (Koller & Pfeffer, 1998; Taskar et al., 2002), Relational Bayesian Classifier (Neville et al., 2003b), Relational Probability Trees[16] (Neville et al., 2003a)). 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 by Popescul et al. (2002) and Naive Bayes with ILP by Pompe and Kononenko (1995)). Krogel and Wrobel (2001) and Knobbe et al. (2001) to our knowledge were 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 (2001) use naive Bayes in combination with FOIL to construct features for hypertext classification.Perlich and Provost (2003) 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 (2004) develop a general Bayesian framework that is closely related to our analysis in Section 3.4, but apply it only to normal attributes with limited cardinality. Although they do not learn from identifiers, Slattery and Mitchell 2000) present a method that takes advantage of the identity of hub pages in a test set (and the fact that they point to many objects of the same class). This can be viewed as a form of collective inference (Jensen et al., 2004) which also (implicitly) reasons based on information about particular nodes.

Theoretical work outside of relational modeling investigates the extension of relational algebra (Özsoyoğlu et al., 1987) through aggregation; however it does not suggest new operators. Libkin and L. Wong (1994) 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, Michalski, 1983) 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 (1993) extended the definition

[16] Relational Probability Trees are an example of a technique that uses simple aggregation operators, but chooses them dynamically in the context of building a specific tree.

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 Kietz and Morik (1994), Morik (1999), and Kramer et al. (2001) for logic-based approaches.

6. Conclusion

The empirical analysis shows that ACORA's distribution-based aggregators facilitate learning (using traditional logistic regression) in domains where to-be-classified entities are linked to other entities. As we highlight in the analysis, ACORA can excel even when the *only* information available for classification is the collection of identifiers of linked entities. Most existing relational learning systems are not equipped to handle object identifiers. ACORA computes class-conditional distributions of linked object identifiers (or other attribute values), and for each to-be-classified instance creates new features by computing distances from these distributions to the particular values linked to the instance.

These new features satisfy the design principles laid out in Section 2: they discriminate instances well, because they yield fine-grained (continuous) values; they induce a similarity with respect to the task, because they are based on class-conditional distributions, and they can capture different notions of similarity by using different distance measures. The results also show that the choice of aggregation can have a significant effect on modeling ability. In many of the domains in the analysis, traditional aggregations simply were not applicable. Other distribution-based aggregators used by ACORA, such as binary features indicating a connection to (i) a very common individual object or (ii) a highly discriminative individual object, also facilitate learning in these relational domains. However, generally they are not as effective as the distributional distances because they do not satisfy the design principles as well—(ii) does not discriminate instances as well, and (i) in addition does not focus on the classification task.

As suggested by the theoretical development in Section 3.5, the empirical results show that aggregated object identifiers can: proxy for important hidden variables, proxy for information deeper in the relational network, and deal with violations of conditional independence assumptions among attributes. As a practical example, a wireless call from a defrauded account may be placed from a particular "cell site," because that cell site is the locus of criminal activity (Fawcett & Provost, 1997). This attribute of the location may not be known a priori; however, if defrauded accounts previously called from the cell site, then the cell site's identity may be sufficient for inference (especially when aggregated with other such locations). An important area for future work is to understand the pros and cons of modeling with identifiers as compared to explicitly trying to model (latent) groups (Neville & Jensen, 2005).

ACORA's aggregators nonetheless have limitations that should be the subject of future research. As explained by the theoretical development in Section 3.4, these aggregators make several layers of independence assumptions. For example, they assume independence among the attributes of related entities. This leads to potentially important concepts that they cannot capture, for example: purchasing different (distributions of) products on different days of the week; calling different numbers at different times of the day. Perlich and Provost (2003) discuss a hierarchy of levels of complexity of aggregation for relational learning. To our knowledge, only inductive logic programming systems address aggregations that take into account important dependencies among attributes of related instances, and as described above, their aggregation usually is quite limited.

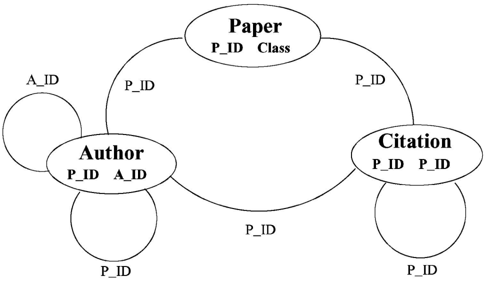

Fig. 11 Graph representation of the CORA document classification domain, with target table Paper(P_ID and Class), and two background tables Author(P_ID,A_ID) and Citation(P_ID,P_ID). Each identifier also produces a self-loop, except on the target table

Another limitation of the features constructed by ACORA is that they are not easily comprehensible. The only conclusion that could be drawn about the use by a model of a vector-distance feature is that the distribution of values of the 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 DV^0 and DV^1.

The distribution-based approach to aggregation is not limited to categorical values. It applies easily to numeric variables if one makes strong distributional assumptions (e.g., Normality). Via discretization it also can be applied to numeric attributes with arbitrary distributions. 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 (cf., Popescul & Ungar, 2004).

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 useful aggregation methods.

Appendix A: 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 heuristic proposed above that requires an identifier to be categorical and to have cardinality larger than some constant, which we 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 (McCallum et al., 2000), a bibliographic database of machine learning papers (see Section 6). CORA comprises three tables (Paper, Author and Citation) as described in Table 5. 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 Q and R represents the occurrence of a pair of identifier attributes $Q.l$ and $R.k$ that are compatible, i.e., they belong to the equality relation $ER(Q.l, R.k)$. The only condition imposed on an edge is that Q and R cannot both be the target table T This allows for multiple edges between two tables. With the exception of the target table,

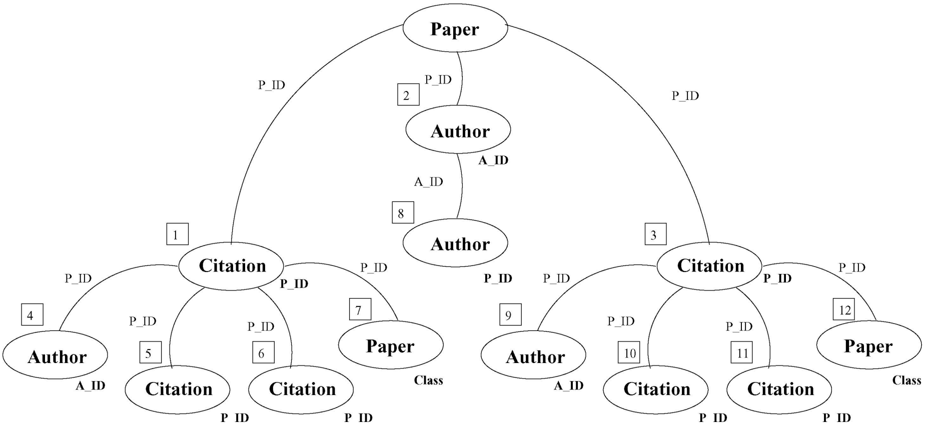

Fig. 12 Search tree corresponding to a breadth-first search over the CORA graph in Figure 11. The exploration for bags of related objects starts from the target table Paper. The numbers denote the order in which the nodes are visited; attribute names on links show the identifier that was used for the join, and the attribute names to the right of each node denote attributes that have to be aggregated

we also allow edges that link a table to itself.[17] Figure 11 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 the fields 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 12 shows the "unrolled" search tree; the numbers correspond 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 a layer of depth *n* represent all possible 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 to the citation table produces a bag of all cited papers. 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*. Tables resulting from a path that reuses the same key only result in a replication of information that is 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 12) 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 usually decreases. Alternative stopping criteria include the number of constructed features, run time, minimum gain in model performance, etc. Finally note that linking back to the

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

target table may or may not be desirable (Provost et al., 2003); doing so often is appropriate for networked domains (Chakrabarti et al., 1998; Domingos & Richardson, 2001; Macskassy & Provost, 2003).

Appendix B: 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 4.4) 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, in the main results we do not include the values of x and y but provide only the identifier.

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 (Fawcett & Provost, 1997; Cortes et al., 2002); there are many variants, but generally speaking accounts are flagged as suspicious if they call numbers in F.

We generate 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.

Customer behavior (KDD)

Blue Martini (Zheng et al., 2001) 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 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 (Bernstein et al., 2002). 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 typically are 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 (McCallum et al., 2000) 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 11. 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 10.

Acknowledgment We are grateful to William Greene, David Jensen, Sofus Macskassy, Jennifer Neville, 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 grateful to Hendrick Blockeel 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.

References

Bernstein, A., Clearwater, S., Hill, S., Perlich, C., & Provost, F. (2002). Discovering knowledge from relational data extracted from business news. In *Proceedings of the Workshop on Multi-Relational Data Mining at KDD-2002* (pp. 7–20). University of Alberta, Edmonton, Canada.

Blockeel, H., & Raedt, L.D. (1998). Top-down induction of first-order logical decision trees. *Artificial Intelligence*, *101*, 285–297.

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

Brazdil, P., Gama, J., & Henery, R. (1994). Characterizing the applicability of classification algorithms using meta level learning. In *Proceedings of the 7th European Conference on Machine Learning* (pp. 83–102).

Chakrabarti, S., Dom, B., & Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. In *Proceedings of the International Conference on Management of Data* (pp. 307–318).

Cortes, C., Pregibon, D., & Volinsky, C. (2002). Communities of interest. *Intelligent Data Analysis, 6:3*, 211–219.

Craven, M., & Slattery, S. (2001). Relational learning with statistical predicate invention: Better models for hypertext. *Machine Learning*, *43*, 97–119.

DerSimonian, R., & Laird, N. (1986). Meta-analysis in clinical trials. *Controlled Clinical Trials*, *7*, 177–188.

Domingos, P., & Richardson, M. (2001). Mining the network value of customers. In *Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining* (pp. 57–66).

Fawcett, T., & Provost, F. (1997). Adaptive fraud detection. *Data Mining and Knowledge Discovery*, *1*, 291–316.

Flach, P., & Lachiche, N. (2004). Naive Bayesian classification for structured data. In *Machine Learning*, *57*, 233–269.

Gärtner, T. (2003). A survey of kernels for structured data. *SIGKDD Explorations, 5*, 49–58.

Gärtner, T., Lloyd, J.W., & Flach, P.A. (2002). Kernels for structured data. In *Proceedings of the 12th International Conference on Inductive Logic Programming* (pp. 66–83). Springer.

Goldberg, H., & Senator, T. (1995). Restructuring databases for knowledge discovery by consolidation and link formation. In *Proceedings of the 1st International Conference On Knowledge Discovery and Data Mining* (pp. 136–141). Montreal, Canada: AAAI Press.

Jensen, D., & Getoor, L. (2003). In *Proceedings of the Workshop on Learning Statistical Models from Relational Data at IJCAI-2003*. American Association for Artificial Intelligence.

Jensen, D., & Neville, J. (2002). Linkage and autocorrelation cause feature selection bias in relational learning. In *Proceedings of the 19th International Conference on Machine Learning* (pp. 259–266). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Jensen, D., Neville, J., & Gallagher, B. (2004). Why collective inference improves relational classification. In *Proceedings of the 10th International Conference on Knowledge Discovery and Data Mining* (pp. 593–598). New York, NY, USA: ACM Press.

Jensen, D., Neville, J., & Hay, M. (2003). Avoiding bias when aggregating relational data with degree disparity. In *Proceedings of the 20th International Conference on Machine Learning* (pp. 274–281).

Kietz, J.-U., & Morik, K. (1994). A polynomial approach to the constructive induction of structural knowledge. *Machine Learning, 14*, 193–217.

Kirsten, M., Wrobel, S., & Horvath, T. (2000). Distance based approaches to relational learning and clustering. In S. Ďzeroski & N.Lavrač (Eds.), *Relational data mining*, (pp. 213–232). Springer Verlag.

Knobbe, A., Haas, M.D., & Siebes, A. (2001). Propositionalisation and aggregates. In *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery* (pp. 277–288).

Koller, D., & Pfeffer, A. (1998). Probabilistic frame-based systems. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI)* (pp. 580–587).

Kramer, S., Lavrač, N., & Flach, P. (2001). Propositionalization approaches to relational data mining. In S. Ďzeroski and N. Lavrač (Eds.), *Relational data mining* (pp. 262–291). Springer-Verlag.

Krogel, M.-A., Rawles, S., Železng, F., Flach, P., Lavrač, N., & Wrobel, S. (2003). Comparative evaluation of approaches to propositionalization. In *13th International Conference on Inductive Logic Programming (ILP)* (pp. 197–214).

Krogel, M.-A., & Wrobel, S. (2001). Transformation-based learning using multirelational aggregation. In *Proceedings of the 11th International Conference on Inductive Logic Programming (ILP)* (pp. 142–155).

Krogel, M.-A., & Wrobel, S. (2003). Facets of aggregation approaches to propositionalization. In *Proceedings of the 13th International Conference on Inductive Logic Programming (ILP)* (pp. 30–39).

Lavrač, N., & Ďzeroski, S. (1994). *Inductive logic programming: techniques and application*. New York. Ellis Horwood, Libkin, L., & Wong L. (1994). New techniques for studying set languages, bag languages and aggregate functions. In *Proceedings of the 13th Symposium on Principles of Database Systems* (pp. 155–166).

Macskassy, S., & Provost, F. (2003). A simple relational classifier. In *Proceedings of the Workshop on Multi-Relational Data Mining at KDD-2003.*

Macskassy, S., & Provost, F. (2004). *Classification in networked Data: A Toolkit and a Univariate Case Study* (Technical Report CeDER-04-08). Stern School of Business, New York University.

McCallum, A., Nigam, K., J. Rennie, & Seymore, K. (2000). Automating the construction of internet portals with machine learning. *Information Retrival, 3*, 127–163.

McCreath, E. (1999). *Induction in First Order Logic from Noisy Training Examples and Fixed Example Set Size*. Doctoral dissertation, Universtity of New South Wales.

Michalski, R. (1983). A theory and methodology of inductive learning. *Artificial Intelligence, 20*, 111–161.

Morik, K. (1999). Tailoring representations to different requirements. In *Proceedings of the 10th International Conference on Algorithmic Learning Theory (ALT)* (pp. 1–12).

Muggleton, S. (2001). CProgol4.4: A tutorial introduction. In S. Ďzeroski & N.Lavrač (Eds.), *Relational Data Mining* pp.(105–139). Springer-Verlag.

Muggleton, S., & DeRaedt, L. (1994). Inductive logic programming: Theory and methods. *The Journal of Logic Programming, 19 & 20*, 629–680.

Neville, J., & Jensen, D. (2005). Leveraging relational autocorrelation with latent group models. In *Proceedings of the 5th IEEE International Conference on Data Mining* (pp. 49–55). New York, NY, USA: ACM Press.

Neville, J., Jensen, D., Friedland, L., & Hay, M. (2003a). Learning relational probability trees. In *Proceedings of the 9th International Conference on Knowledge Discovery and Data Mining* (pp. 625–630). New York, NY, USA: ACM Press.

Neville, J., Jensen, D., & Gallagher, B. (2003b). Simple estimators for relational Bayesian classifers. In *Proceedings of the 3rd International Conference on Data Mining* (pp. 609–612). Washington, DC, USA: IEEE Computer Society.

Neville, J., Rattigan, M., & Jensen, D. (2003c). Statistical relational learning: Four claims and a survey. In *Proceedings of the Workshop on Learning Statistical Models from Relational Data at IJCAI-2003.*

Özsoyoğlu, G., Özsoyoğlu, Z., & Matos, V. (1987). Extending relational algebra and relational calculus with set-valued atributes and aggregate functions. *ACM Transactions on Database Systems, 12*, 566–592.

Perlich, C. (2005a). Approaching the ILP challenge 2005: Class-conditional Bayesian propositionalization for genetic classification. In *Late-Braking track at the 15th International Conference on Inductive Logic Programming* (pp. 99–104).

Perlich, C. (2005b). *Probability estimation in mulit-relational domain.* Doctoral dissertation, Stern School of Business.

Perlich, C., & Provost, F. (2003). Aggregation-based feature invention and relational concept classes. In *Proceedings of the 9th International Conference on Knowledge Discovery and Data Mining* (pp. 167–176). New York, NY, USA: ACM Press.

Perlich, C., Provost, F., & Simonoff, J. (2003). Tree induction vs. logistic regression: A learning-curve analysis. *Journal of Machine Learning Research, 4*, 211–255.

Pompe, U., & Kononenko, I. (1995). Naive Bayesian classifier with ILP-R. In *Proceedings of the 5th International Workshop on Inductive Logic Programming* (pp. 417–436).

Popescul, A., & Ungar, L. (2003). Structural logistic regression for link analysis. In *Proceedings of the Workshop on Multi-Relational Data Mining at KDD-2003.*

Popescul, A., & Ungar, L. (2004). Cluster-based concept invention for statistical relational learning. In *Proceedings of the 10th International Conference on Knowledge Discovery and Data Mining* (pp. 665–670).

Popescul, A., Ungar, L., Lawrence, S., & Pennock, D.M. (2002). Structural logistic regression: Combining relational and statistical learning. In *Proceedings of the Workshop on Multi-Relational Data Mining at KDD-2003* (pp. 130–141).

Provost, F., Perlich, C., & Macskassy, S. (2003). Relational learning problems and simple models. In *Proceedings of the Workshop on Learning Statistical Models from Relational Data at IJCAI-2003* (pp. 116–120).

Quinlan, J. (1993). *C4.5: Programs for machine learning.* Los Altos, California: Morgan Kaufmann Publishers.

Quinlan, J., & Cameron-Jones, R. (1993). FOIL: A midterm report. In *Proceedings of the 6th European Conference on Machine Learning (ECML)* (pp. 3–20).

Slattery, S., & Mitchell, T. (2000). Discovering test set regularities in relational domains. In *Proceedings of the 17th International Conference on Machine Learning* (pp. 895–902).

Taskar, B., Abbeel, P., & Koller, D. (2002). Discriminative probabilistic models for relational data. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence* (pp. 485–492). Edmonton, Canada: Morgan Kaufmann.

Taskar, B., Segal, E., & Koller, D. (2001). Probabilistic classification and clustering in relational data. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence* (pp. 870–878).

Witten, I., & Frank, E. (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

Wnek, J., & Michalski, R. (1993). Hypothesis-driven constructive induction in AQ17-HCI: A method and experiments. *Machine Learning, 14*, 139–168.

Woznica, A., Kalousis, A., & Hilario, M. (2004). Kernel-based distances for relational learning. In *Proceedings of the Workshop on Multi-Relational Data Mining at KDD-2004*.

Zheng, Z., Kohavi, R., & Mason, L. (2001). Real World Performance of Association Rule Algorithms. In *Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining* (pp. 401–406).