Noname manuscript No.
(will be inserted by the editor)

Distribution-free Bounds for Relational Classification

Amit Dhurandhar · Alin Dobra

Received: Mar 22, 2010 / Revised: Mar 29, 2011 / Accepted: Apr 16, 2011

Abstract Statistical Relational Learning (SRL) is a sub-area in Machine Learning which addresses the problem of performing statistical inference on data that is correlated and not independently and identically distributed (i.i.d.) – as is generally assumed. For the traditional i.i.d. setting, distribution free bounds exist, such as the Hoeffding bound, which are used to provide confidence bounds on the generalization error of a classification algorithm given its hold-out error on a sample size of N. Bounds of this form are currently not present for the type of interactions that are considered in the data by relational classification algorithms. In this paper we extend the Hoeffding bounds to the relational setting. In particular, we derive distribution free bounds for certain classes of data generation models that do not produce i.i.d. data and are based on the type of interactions that are considered by relational classification algorithms that have been developed in SRL. We conduct empirical studies on synthetic and real data which show that these data generation models are indeed realistic and the derived bounds are tight enough for practical use.

Keywords data mining, relational learning, bounds, classification

1 Introduction

Statistical Relational Learning (SRL) [13] deals with modeling uncertainty in relational data. The primary objective of this sub-area of Machine Learning/Data Mining is to move away from the independent and identically distributed (i.i.d.) assumption that is omnipresent in traditional Machine Learning and to start modeling dependencies between related data instances. One of the key SRL tasks is collective classification. In collective classification related data instances are classified simultaneously rather than independently as done in traditional classification. Though there are numerous

Amit Dhurandhar
IBM T.J. Watson
E-mail: adhuran@us.ibm.com

Alin Dobra
University of Florida
E-mail: adobra@cise.ufl.edu

relational classification algorithms [34, 38, 12, 26, 24] developed in literature, the current state of theory – distribution free bounds, for relational domains is still in its infancy. The need for deriving such bounds for the relational setting has been expressed in [17, 13].

In Machine Learning, distribution free bounds are used to bound the empirical error (test or training error) of a classifier with respect to (w.r.t.) its generalization error. The generalization error of a classifier is defined as the expected error of a classifier on the entire input. The expectation is w.r.t. the underlying joint distribution of the available data or sample. The generalization error is thus the true error of a classifier. Knowing the generalization error is an invaluable piece of information as it helps us evaluate a particular classifier and appropriately choose the best classifier if multiple options are available. In reality though, this error cannot be computed directly, since the underlying joint distribution of the sample is practically never known. The empirical error on the other hand can be computed from the sample. Distribution free bounds relate these two errors by providing us with probabilistic estimates for the generalization error given the empirical error without knowledge of the underlying joint distribution. This is the primary reason for deriving such bounds [16, 39, 10, 6, 23, 22]. A common characteristic of all of these bounds is that they assume the data is i.i.d. This is a strong assumption on the underlying distribution or the data generation process as it ignores interactions that may exist between datapoints. In this paper we forgo the i.i.d. assumption and derive Hoeffding style bounds [16] on the generalization error in the presence of interactions between data instances.

1.1 Specific Contributions

The specific contributions we make in this paper are:

1. We derive distribution free bounds for the generalization error of classification algorithms, where the data generation models produce non-i.i.d. data. The first and a key step in deriving such bounds is to characterize the data generation process. We define 2 classes of data generation models namely; C1 and C2 which consider fixed size interactions and variable size interactions respectively. We motivate and elaborate on these models in the next couple of sections.
2. We explain what the bounds convey and their relation to effective sample size [18]. In [18], the authors introduced this notion and empirically observed it for the relational setting. Particular terms in the bounds that we derive for the generalization error, can be interpreted as lower bounds on this quantity. Though the primary goal of this work is to obtain bounds on the generalization error in the relational setting, the relation to effective sample size is a direct consequence of the derived formulae.
3. To obtain an intuitive feel for the behavior of the derived bounds, we empirically evaluate them w.r.t. increasing size of the interactions, increasing dataset size N and increasing number of interactions. We also show how the derived bounds are applicable in real settings. We accomplish this by conducting experiments on synthetic and real data where we train a state-of-the-art relational classification algorithm and then apply the derived bounds to the estimated error.

The remainder of the paper is organized as follows: In Section 2 we explain certain basic concepts and motivate the data generation models considered in this paper. In

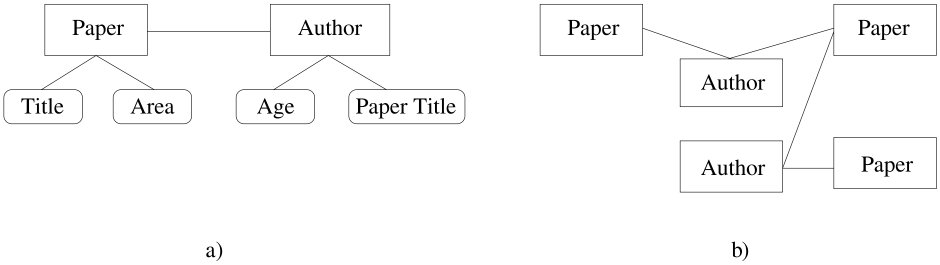

Fig. 1 a) represents a relational schema with object types, *Paper* and *Author*. The relationship between them is many-to-many. The rounded boxes linked to these object types denote their respective attributes. b) is the corresponding data graph which shows authors linked to the papers that they authored or co-authored.

Section 3 we explain why we chose the Hoeffding bound to be extended to the relational setting. In Section 4 we formally define the class of data generation models namely; C1 and C2. In Section 5 we review related work. In Section 6 we initially provide some previously known results, then state the derived inequalities in Lemmas 1 and 2 and provide proofs for the Lemmas 1 and 2. In Subsection 6.3 we explain the semantics of the derived inequalities and state their relation to effective sample size. In Section 7 we empirically evaluate the bounds on synthetic and real data. In Section 8 we mainly discuss strategies to derive tight bounds. We conclude in Section 9 wherein we summarize the major findings in this paper.

2 Preliminaries and Motivation

As we mentioned in the introduction, a key step in deriving distribution free bounds is to characterize the type of interactions between interacting datapoints which in turn will determine the structure of the data generation models. To motivate the data generation models that we consider in this paper, we first review the characteristics of relational data and discuss the type of probability distributions learned over this data by state-of-the-art relational classification algorithms. These distributions motivate the data generation models for which we derive bounds.

Relational data: Relational data consists of objects and the relationships between these objects are termed as links. Each object and link have a *type* associated with them. Objects or links of the same *type* have the same set of attributes. Relational data can be represented at the *type* level by a graph which is called a relational schema whereas relational data represented at the individual object and link level as a graph is called a relational data graph (or instance graph) [24], wherein the vertices are the objects and the edges are the links. An example relational schema and the corresponding data graph (i.e. the actual dataset) are shown in Figures 1a and 1b respectively. The relational schema has 2 object types namely; *Paper* and *Author*. The data graph shows 2 authors linked to the papers they authored or co-authored.

Probabilistic Models over Relational Data: Probabilistic Models over relational data (PMRD) [13] are structured graphical models that are used to handle uncertainity in relational domains. A PMRD represents a joint distribution over the attributes of a

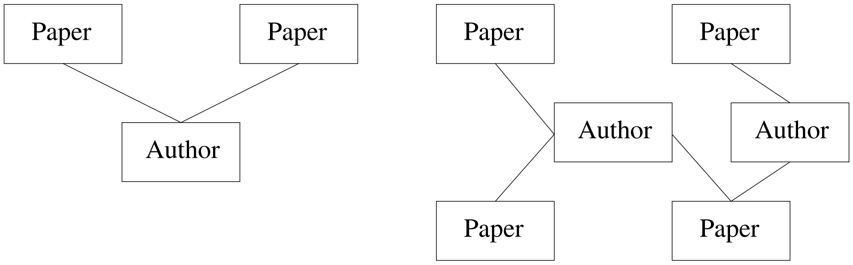

Fig. 2 Above we see a disconnected data graph with 2 components of variable size (i.e. size 2 component on the left and size 4 component on the right).

data graph. Consider Figure 1a where the object type *Paper* has 2 attributes, Title and Area which imply the title of the paper and the research area it belongs to respectively. Let the attribute Area be the class label i.e. we want to classify papers based on their research area. The object type *Author* has attributes Paper Title and Age, which relates a particular paper to the ages of the authors that wrote it. The Title attribute (a primary key) in *Paper* is the same as the Paper Title attribute (a foreign key) in *Author*. Hence, each Paper object has 3 attributes namely; Title, Area and Age. The attributes Title and Area are called *intrinsic attributes* as they belong to object type *Paper* and the attribute Age is called a *relational attribute* since it belongs to a different linked object type *Author*. Each paper can have variable number of authors and thus each paper would be associated with multiple values of Age. A popular solution to this problem is to aggregate the values of the attribute Age of *Author* into a single value such that each paper is associated with only a single Age value. An aggregation function such as average over the ages of the related authors for each paper can be used. Now instead of the Age attribute we can introduce a new attribute AvgAge which denotes average age. With this the attributes of Paper object are; Title, Area and AvgAge. Hence, the joint distribution represented by a PMRD on the data graph in Figure 1b is,

$$P[Title^1, Area^1, AvgAge^1, Title^2, Area^2, AvgAge^2, Title^3, Area^3, AvgAge^3]$$

where the superscripts $\{1, 2, 3\}$ denote the corresponding Paper objects. Since in Figure 1b we have paths connecting the 3 papers (through authors), we have 3 copies of the same attributes (which may have different values) in the joint distribution.

The data graph in Figure 1b is connected. It is possible in some other case that the data graph is actually disconnected. This is shown in Figure 2. Let A_i denote the attribute set $\{Title^i, Area^i, AvgAge^i\}$ of the i^{th} Paper object. The joint distribution over the data graph in Figure 2 is,

$$P[A_1, A_2, ..., A_6] = P[A_1, A_2]P[A_3, A_4, A_5, A_6]$$

Since the data graph has 6 Paper objects, we have 6 copies of the attributes. Moreover, there are 2 disconnected components in the graph, one with 2 Paper objects and the other with 4 Paper objects. Consequently, the joint probability distribution $P[A_1, A_2, ..., A_6]$ factorizes as a product of 2 independent distributions $P[A_1, A_2]$ and $P[A_3, A_4, A_5, A_6]$. Notice that in the extreme case if we had all 6 Paper objects disconnected the data would be treated by a PMRD as *i.i.d.* since there would be no

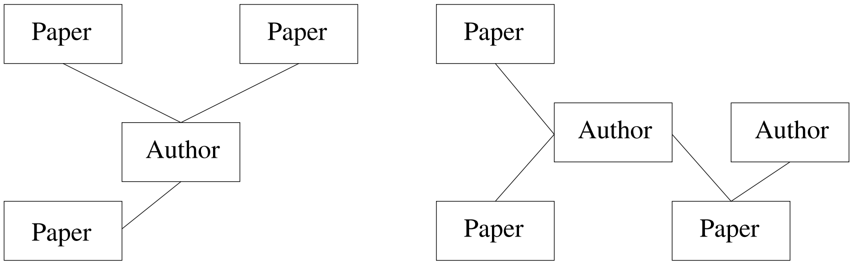

Fig. 3 Above we see a disconnected data graph with 2 components of size 3 (since each component has 3 Paper objects).

interactions between any two or more papers. In this case the joint distribution would factorize as follows,

$$P[A_1, A_2, ..., A_6] = P[A_1]P[A_2]P[A_3]P[A_4]P[A_5]P[A_6]$$

Since each of the 6 factorized probabilities have the same set of attributes the probability distribution determined by them are the same and the distribution for such a setting is completely characterized by any single $P[A_i]$ ($i \in \{1, ..., 6\}$). Extending this idea to the case where we have multiple components of the same size, the distribution would be completely characterized by the probability distribution over the attributes of any single component. For example in Figure 3 the data graph has 2 components of size 3 (since each has 3 Paper objects). The joint distribution over this data graph is given by:

$$P[A_1, A_2, ..., A_6] = P[A_1, A_2, A_3]P[A_4, A_5, A_6]$$

but is completely characterized by $P[A_1, A_2, A_3]$ *or* $P[A_4, A_5, A_6]$, since both have the same set of attributes (i.e. are over the same space) and represent the same distribution.

We have thus observed that joint distributions over relational data can be over a single set of interactions (i.e. one component connected graph) which results in a single joint probability over all the attributes or multiple sets of independent interactions (i.e. multiple component disconnected graph) which results in factorization of the joint probability into independent distributions. If the independent sets of interactions are of the same size (called fixed size interactions), the joint probability is completely characterized by any of the independent distributions. We consider the scenario wherein we have fixed size interactions separate from the more general scenario wherein we have interactions are of arbitrary size, since the ideas used to derive bounds for the data generation models in the first scenario make it easier to follow the proof for the bounds derived for the data generation models in the second scenario. Moreover, certain datasets may have fixed size interactions in which case the bound derived for this case can be directly applied. A major portion of the rest of the paper is devoted to deriving bounds for joint distributions or data generation models belonging to these 2 scenarios.

3 Why extend the Hoeffding Bound?

In [22] there is an extensive review of different types of distribution free bounds that are prevalent in Machine Learning. The author categorizes each bound into one of

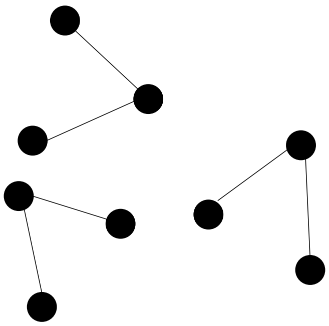

Fig. 4 Class C1: Fixed size interactions between datapoints of size 3.

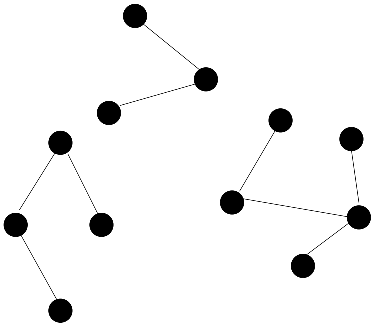

Fig. 5 Class C2: Variable size interactions between datapoints of sizes 3, 4, 5.

the two categories namely; as a test set bound or as a training set bound[1]. As the name suggests test set bounds bound the error over the test set by considering that the error has a binomial distribution. A good approximation to this bound in the agnostic setting is the Chernoff bound [8] and the related Hoeffding bound. A training set bound on the other hand bounds the training error w.r.t. the generalization error. Well known examples of this type of bound are the Vapnik-Chervonenkis bounds (VC bounds) [39], Probably Approximately Correct Bayes bounds (PAC Bayes bounds) [23], Occam's Razor bounds [7], Sample Compression bounds [11] and Rademacher Complexity bounds [4]. The author infers from the comparison of these two categories that test set bounds are generally much tighter than training set bounds and are a superior tool in reporting error rates. The derivation of the test set bound however, requires that the probability of errors on individual inputs be independent and identical to each other. This assumption is central to the proof of these bounds and seems almost impossible to relax. Hence, the natural alternative that we can aim at extending to the relational setting are the approximations to these bounds which are the Chernoff and Hoeffding bounds.

In this paper we extend the Hoeffding bound to the relational setting. Another advantage of extending this bound besides the one mentioned above is that the amount of information required to compute it is minimal and simple to obtain – N the number of i.i.d. random variables (i.e. test set size) and the range of the random variables (generally $[0, 1]$) – as opposed to the other bounds (for example Chernoff bound requires computation of Kullback-Leibler divergence). Moreover, it decreases exponentially with increasing N which makes it tighter than other similar bounds [16]. Considering all of these facts extending the Hoeffding bound seems to be a reasonable initial step in deriving useful distribution free bounds for relational data.

4 Data Generation Models

In section 2 we observed the type of joint probability distributions or data generation models that a PMRD aims to characterize over a relational data graph. It was seen

[1] Some bounds lie in both categories and can be useful in specific circumstances.

that a relational data graph provides the underlying structure to the joint probability distribution represented by a PMRD. The goal of the paper is to derive Hoeffding style inequalities for such data generation models. The resulting inequalities will bound the appropriately defined test error w.r.t. the generalization error of the classifier in question. The bounds however, depend on the structure of the underlying probabilistic space which needs to be characterized first, if we are to derive them. Hence, we now formally define the data generation models that characterize the underlying probabilistic space and derive the relevant bounds later. The data generation models we consider are motivated from models seen in section 2.

Given N datapoints (i.e. objects. Paper object in our example above) in a d dimensional space (this includes intrinsic as well as relational attributes) and considering that all the N points interact with each other (i.e. one connected component in the data graph), the resultant data generation process which accurately models this scenario has the following form, $P[X_1^1, ..., X_d^1, ..., X_1^N, ..., X_d^N]$ where X_i^j is the j^{th} copy of the i^{th} attribute. This is the full-fledged data generation model wherein every datapoint interacts with every other datapoint. The parameter space of this model is $O(v^{Nd})$ as opposed to $O(v^d)$ in the usual i.i.d. setting where say v is the number of distinct values of each attribute. Thus, the full-fledged data generation model and the i.i.d. model occupy two ends of a spectrum consisting of other generation models which consider limited (i.e. interactions *not* between all datapoints but between smaller partitions) but non-trivial (i.e. there is atleast one pair of datapoints which are dependent) interactions. With this we now define two classes of data generation models which include models in this spectrum. We refer to the models in class 1 as C1 models and analogously those in class 2 as C2 models.

1. *C1 models:* The C1 data generation models consider interactions of fixed size. This means that the joint distributions considered by models in this class are over data graphs that have one or more components with the same number objects of the object type that is to be classified.
   For example, the distribution over the data graph in Figure 1b wherein 3 papers are linked belongs to this class. If the data graph had multiple components of 3 papers linked through authors as in Figure 3 then the distribution over this data graph also belongs to this class since the overall joint distribution factorizes into independent distributions with the same attributes making them identical to each other. On the other hand the distribution over the data graph in Figure 2 does not belong to this class since there exist variable size interactions (i.e. one component with 2 Paper objects linked and another with 4 Paper objects linked). An example of sets of 3 datapoints interacting with each other is shown in Figure 4.
   The data generation model representing this scenario has the following form,
   $P[X_1^1, ..., X_d^1, ..., X_1^3, ..., X_d^3]$. Notice that a set of 3 points in the original d dimensional space is a single sample from this $3d$ parameter distribution. Moreover, each such sample is i.i.d. from this new distribution. In the general case the 3 can be any number $1 \leq m \leq N$ with the data generation model having the following form, $P[X_1^1, ..., X_d^1, ..., X_1^m, ..., X_d^m]$.
2. *C2 models:* The next type data generation models we consider are more general than the above model. Here we consider interactions of arbitrary size. The joint distribution over the data graph in Figure 2 belongs to this class. Thus, we can have a set of 5 points interacting, a set of 4 points interacting and a set of 3 points

interacting as can be seen in Figure 5. The data generation model has the following form,

$$\begin{aligned}&P[X_1^1,...,X_d^1,...,X_1^N,...,X_d^N] =\\ &\quad P[X_1^1,...,X_d^1,...,X_1^j,...,X_d^j]\cdot\\ &\quad P[X_1^{j+1},...,X_d^{j+1},...,X_1^i,...,X_d^i]\cdot\\ &\quad ...\cdot P[X_1^k,...,X_d^k,...,X_1^N,...,X_d^N]\end{aligned}$$

where $1 \leq j < i < k$ and $i, j, k \leq N$.

Notice that the i.i.d. data generation model and the full-fledged model both lie in C1 and C2.

4.1 Assumption

An important thing to note is that considering the structure of our problem where we have copies of the same attribute occuring as many times as the size of the interactions, the marginal probabilities over the attributes of the interacting datapoints can be assumed to be equal. In other words, we have copies of the same attribute interacting with each other and hence, if we consider each of these copies in isolation, they should have the same type of behavior. Formally, the data generation process $P[X_1^1,...,X_d^1,...,X_1^N,...,X_d^N]$ where X_i^j is the j^{th} copy of the i^{th} attribute, can be factorized in the following possible ways (not an exhaustive list),

$$\begin{aligned}P[A_1,...,A_N] &= P[A_1]P[A_2|A_1]...P[A_N|A_{N-1},...,A_1]\\ &= P[A_2]P[A_1|A_2]...P[A_N|A_{N-1},...,A_1]\\ &.\\ &.\\ &.\\ &= P[A_N]P[A_1|A_N]...P[A_{N-1}|A_N,A_{N-2},...,A_1]\end{aligned}$$

where A_i (input-output space of the i^{th} datapoint or the i^{th} copy of the attributes) denotes the set $\{X_1^i,...,X_d^i\}$ ($i \in \{1,...,N\}$). If we sample from the above joint distribution using the factorization, the first sample would either be drawn from $P[A_1]$ (if the first factorization is used) or $P[A_2]$ (if the second factorization is used) or some other $P[A_j]$ (if the j^{th} factorization is used) depending on the factorization used. The probability of the first sample in the original d dimensional space should be the same irrespective of which set of copies of attributes are used to obtain it. Consequently, we make the following assumption,

$$P[A_1] = P[A_2] = ... = P[A_N]$$

Note that the above assumption does *not* imply that the data is i.i.d. since the data generation process is given by the joint distribution $P[A_1,...,A_N]$ with interactions between various A_i ($i \in \{1,...,N\}$). The assumption only implies that the respective marginals are equal. In other words, the data being i.i.d. $\Rightarrow P[A_1] = P[A_2] = ... =$

$P[A_N]$ but $P[A_1] = P[A_2] = ... = P[A_N]$ *does not* $\Rightarrow$ that the data is i.i.d. since the joint probability may not factorize as a product of the individual probabilities.

For example in Figure 1b where the joint distribution over the data graph is given by: $P[Title^1, Area^1, AvgAge^1, Title^2, Area^2, AvgAge^2, Title^3, Area^3, AvgAge^3]$ where the superscripts $\{1, 2, 3\}$ denote the corresponding copies of the attributes of Paper, we assume that $P[Title^1, Area^1, AvgAge^1] = P[Title^2, Area^2, AvgAge^2] = P[Title^3, Area^3, AvgAge^3]$ which certainly does not imply that the respective copies of attributes are independent w.r.t. each other.

In fact the above assumption is implicitly made by a PMRD since it learns at the type level [13]. We will use the above result in deriving the bounds.

5 Related Work

Distribution free bounds other than those given by Hoeffding exist in literature. Amongst the more popular is the Markov inequality [30,15] and the Chebyshev inequality [30, 15] which bound a random variable to its mean. The Hoeffding inequality though, gives tighter bounds than these two inequalities in most cases [16]. Other such inequalities are given by Chernoff [8], Bennett [5], Okamoto [29]. Comparison of these inequalities is given in [14], [36] and [5]. Distribution free bounds on the generalization error of a classifier are provided by Vapnik [39]. In [10] distribution free bounds are provided for the k-nearest neighbor algorithm. In [3] improved PAC Bayes bounds are provided for a class of linear classifiers. These bounds are tighter than the ones previously introduced in [23]. In [6] bounds are provided for a validation technique called progressive validation which are tighter than those for hold-out-set validation. The derivation of these bounds uses Hoeffdings inequality thus portraying its widespread use in Machine Learning. A common characteristic of all of these bounds applied to classification algorithms is that they assume the data is i.i.d.

A plethora of learnability results (both positive and negative) have been proven for restricted classes of inductive logic programs [9,32,2,1]. The learnability results are primarily based on two formal models of learning namely; PAC learning and learning from equivalence and membership queries. In [32] non-monotonic inductive logic programs are shown to be efficiently PAC learnable. Non-inductive logic programs are a special class of clausal theories wherein each clause has a finite number of literals of finite size. In [2,1] learnability results are proven for a restricted class of Horn clauses based on the equivalence and membership learning model.

Probabilistic bounds also exist when k-wise independence is assumed between random variables. k-wise independence is a limited notion of independence where any set of k (or fewer) random variables are assumed to be independent from a total of n random variables ($k \leq n$). Bounds assuming k-wise independence are given in [37] which extend the ideas given by Chernoff and Hoeffding. The use of these bounds is in applications where pseudo-randomness is assumed. One example application area is Sketches where binary random variables are assumed to be either 2-wise, 3-wise or 4-wise independent [35]. In this paper though, the type of interactions that we described in the introduction are significantly different from k-wise independence and hence theory needs to be developed for this setting.

6 Deriving Bounds

In this section we derive bounds for the data generation models defined in C1 and C2. We first state the Hoeffding's inequality,

Theorem 1 *If $X_1, X_2, ..., X_n$ are independent and $a_i \leq X_i \leq b_i$ $(i = 1, 2, ..., n)$, then for $t > 0$*

$$P[\bar{X} - \mu \geq t] \leq e^{\frac{-2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}}$$

where $\bar{X} = \frac{\sum_i X_i}{n}$ and $\mu = E[\bar{X}]$. When the X_i's are also identically distributed the above inequality, bounds the difference between the average and mean (μ) of the n i.i.d. random variables. *The primary use of such inequalities is to derive confidence bounds on the mean.*

In the traditional setting in Machine Learning, we assume that the data is drawn i.i.d. from a distribution $P[X, Y]$ where X is the input space and Y the output space. A classifier $\zeta(.)$ takes as input a particular $x \in X$ and outputs a particular class label $y \in Y$. The error function most commonly used to calculate the error made by a classifier on a particular input is the 0-1 loss function which we denote by $\lambda(.,.)$. The 0-1 loss function takes as input two parameters. It outputs a 0 if the two parameters are equal else if the parameters are unequal it outputs a 1. Thus, if the parameters of the loss function are $\zeta(x)$ and the actual label of x is y, then $\lambda(\zeta(x), y) = 0$ when $\zeta(x) = y$ and $\lambda(\zeta(x), y) = 1$ otherwise. The expected value of $\lambda(\zeta(x), y)$ over $X \times Y$ space is defined as the generalization error (GE) of the particular classifier ζ. Formally,

$$GE = E[\lambda(\zeta(x), y)] \tag{1}$$

The empirical error (HE) computed over a test set of size N is given by, $HE = \frac{\sum_{i=1}^{N} \lambda(\zeta(x_i), y_i)}{N}$ where $x_i \in X$ and $y_i \in Y$ and y_i is the label of x_i. Since, the (X,Y) are i.i.d. and $\lambda(.,.)$ is a deterministic function, $\lambda(.,.)$ applied to each $\zeta(x)$ and y are also i.i.d. Moreover, $GE = E[HE]$. This equality is a result of linearity of expectation. Given this, we can apply the Hoeffding inequality to bound the empirical error with the GE. The bound is as follows,

$$P[HE - GE \geq t] \leq e^{-2Nt^2} \tag{2}$$

Notice that $\lambda(.,.)$ takes values only 0 or 1 which leads to the simplification on the right hand side of the inequality.

With this brief review of the Hoeffding inequality and its application to Machine Learning, we are now ready to first state and later derive bounds for the generative models in C1 and C2.

6.1 Results for C1 and C2 Models

Deriving bounds for C1 and C2 models can be useful in bounding the error of classification algorithms on relational data obtained from real life settings. As seen before, the type of distributions encompassed by C1 and C2 models include but are not limited to the standard iid distributions and hence the bounds we derive are generalizations of the Hoeffding bound. Without these generalizations, we cannot directly apply the Hoeffding bound to relational data.

Though the derived bounds are an extension of Hoeffding's inequality, their derivation poses non-trivial technical challenges. In the traditional i.i.d. setting the Hoeffding inequality bounds the empirical error (i.e. the test error) of a classifier with its generalization error (i.e. expected value of empirical error). Since in the relational setting too, we wish to derive bounds w.r.t. the generalization error, we need to come up with an estimator from the sample whose expected value is the generalization error. Though this estimator is the usual empirical error for C1 models, it is different for C2 models. The construction of the estimator for C2 models, is part of the proof in which we derive bounds for these models. Moreover, we have to ensure for these bounds to hold that the individual random variables that sum up to produce these estimators are independent and have the same expected value. This constraint of the random variables being independent and having the same mean is a subtle observation on our part for the Hoeffding inequality used to bound the generalization error to hold and is weaker than the i.i.d. constraint[2] which is generally considered as necessary for the bound to be applicable in Machine Learning. This observation is absolutely essential in the derivation of the bound for C2 models.

Lemma 1 *Assume we have interactions of size m between datapoints. The corresponding data generation model is given by, $P_m = P[X_1, Y_1, ..., X_m, Y_m]$ where X_i is the input space and Y_i is the output space $(1 \leq i \leq m)$ of the i^{th} interacting datapoint (or the i^{th} copy of attributes). Let s samples be drawn from this distribution such that $N = ms$, which is the number of datapoints in the original space. Then given that $P[X_i, Y_i] = P[X_j, Y_j]$ $\forall i, j \in \{1, ..., m\}$ we have,*

$$P[HE - GE \geq t] \leq e^{\frac{-2Nt^2}{m}}$$

where $HE = \frac{\sum_{i=1}^{N} \lambda(\zeta(x_i), y_i)}{N}$ $((x_i, y_i)$ are the N datapoints) and $GE = E[\lambda(\zeta(x), y)]$.

It can be seen that the bound becomes loose for a fixed N as the size of the interaction m between datapoints increases. This is expected since the parameter space of the generating model increases with increasing interaction.

Lemma 2 *Assume that we have r independent interactions of datapoints of size $m_1, m_2, ..., m_r$ such that $m_1 + m_2 + ... + m_r = N$ where N is the total number of datapoints. The corresponding data generation model is given by,*

$$\begin{aligned} &P[X_{11}, Y_{11}, ..., X_{m_r r}, Y_{m_r r}] \\ &= P[X_{11}, Y_{11}, ..., X_{m_1 1}, Y_{m_1 1}] P[X_{12}, Y_{12}, ..., X_{m_2 2}, Y_{m_2 2}] \cdot ... \\ &\cdot P[X_{1r}, Y_{1r}, ..., X_{m_r r}, Y_{m_r r}] \end{aligned}$$

where X_{ij} is the input space and Y_{ij} is the output space of the i^{th} datapoint (or i^{th} copy of attributes) in the j^{th} set of interactions ($j \in \{1, ..., r\}$ and $i \in \{1, ..., m_j\}$). A sample from this distribution produces N datapoints in the original space. Let $T_j = \sum_{i=1}^{m_j} \lambda(\zeta(x_{ij}, y_{ij}))$ where $j \in \{1, ..., r\}$ and $Q_j = k_j T_j$ where $k_j = \frac{l}{m_j}$ and l is the least common multiple of $m_1, m_2, ..., m_r$. Then given that $P[X_{ij}, Y_{ij}] = P[X_{fg}, Y_{fg}]$ $\forall j, g \in \{1, ..., r\}, \forall i \in \{1, ..., m_j\}$ and $\forall f \in \{1, ..., m_g\}$ we have

[2] Higher moments of each of the random variables in the sum may vary

$$P[HE' - GE \geq t] \leq e^{-2rt^2}$$

where $HE' = \frac{1}{lr} \sum_{j=1}^{r} Q_j$ *and* $GE = E[\lambda(\zeta(x), y)]$.

The above inequality can be used to derive confidence bounds on GE since HE' is a function of the sample and can be easily computed. Notice that in the i.i.d. setting the above inequality reduces to equation 2. The remainder of this section is dedicated towards proofs of the above 2 Lemmas.

6.2 Proofs

We start of with the main idea used in proving Lemmas 1 and 2 and then provide detailed proofs for the same.

Central theme: We know that the Hoeffding inequality, bounds $\bar{X}$ with $E[\bar{X}]$. Since, we want to find bounds on the GE of classification algorithms, the main theme in the proofs is finding the appropriate $\bar{X}$ whose expected value is GE i.e. $E[\bar{X}] = GE$. With this basic theme in mind, we now present the proofs for the above 2 Lemmas.

Proof Here is the proof for Lemma 1. Assume that we have interactions of size m i.e exactly sets of m datapoints are correlated. The generative model which captures this scenario is given by, $P_m = P[X_1, Y_1, ..., X_m, Y_m]$ where X_i is the input space and Y_i is the output space ($1 \leq i \leq m$) of the i^{th} interacting datapoint. A sample from this distribution produces m datapoints in the original space. Consider s samples drawn from this distribution (which makes them i.i.d.) such that $ms = N$. We thus have N datapoints. The empirical error (HE) of a classifier on this dataset is then given by,

$$HE = \frac{1}{N} \sum_{j=1}^{s} \sum_{i=1}^{m} \lambda(\zeta(x_{ij}), y_{ij}) \tag{3}$$

where x_{ij} and y_{ij} ($i \in [1, .., m]$ and $j \in [1, ..., s]$) are values of X_i and Y_i respectively, in the j^{th} sample.

Note that $T_j = \lambda(\zeta(x_{1j}), y_{1j}) + ... + \lambda(\zeta(x_{mj}), y_{mj})$ where $j \in \{1, ..., s\}]$ is a deterministic function of the sample $x_{1j}y_{1j}...x_{mj}y_{mj}$ obtained from P_m. Since, the s samples from P_m are i.i.d., any deterministic function applied to each of them is also i.i.d. Hence, the T_j's where $j \in \{1, ..., s\}$ are i.i.d. By grouping terms together in equation 3 we can rewrite HE as,

$$HE = \frac{1}{N} \sum_{j=1}^{s} T_j \tag{4}$$

Remember that the Hoeffding inequality bounds HE w.r.t. $E[HE]$. In the traditional i.i.d. setting the $E[HE] = GE$, consequently we were able to bound HE with GE. Is this true even in our setting ? The answer is affirmative as we show below.

$$E[HE] = \frac{1}{N} E[\sum_{j=1}^{s} \sum_{i=1}^{m} \lambda(\zeta(x_{ij}), y_{ij})]$$

By linearity of expectation we have,

$$E[HE] = \frac{1}{N}\sum_{j=1}^{s}\sum_{i=1}^{m} E[\lambda(\zeta(x_{ij}), y_{ij})]$$

By our assumption that the marginals $P[X_i, Y_i]$ where $i \in \{1, ..., m\}$ are equal to each other it follows that $E[\lambda(\zeta(x_{ij}), y_{ij})]$ are also equal to each other where $i \in \{1, .., m\}, j \in \{1, ..., s\}$. Moreover, $E[\lambda(\zeta(x_{ij}), y_{ij})] = GE\ \forall i \in [1, .., m], j \in [1, ..., s]$ by definition. Thus we have,

$$\begin{aligned} E[HE] &= \frac{ms}{N} GE \\ &= \frac{N}{N} GE \\ &= GE \end{aligned} \tag{5}$$

From equation 4 we know that HE is the average of s i.i.d. random variables T_j (though each T_j is the sum of dependent random variables) lying in the range [0,m] where $j \in \{1, .., s\}$ divided by m. Thus, $mE[HE] = E[\frac{1}{s}\sum_{j=1}^{s} T_j]$. Thus, the Hoeffding inequality in this scenario is given by,

$$P[\frac{1}{s}\sum_{j=1}^{s} T_j - E[\frac{1}{s}\sum_{j=1}^{s} T_j] \geq t] \leq e^{\frac{-2st^2}{m^2}}$$

the above equation can be rewritten as,

$$P[\frac{1}{s}\sum_{j=1}^{s} T_j - mGE \geq t] \leq e^{\frac{-2Nt^2}{m^3}}$$

The above equation is sufficient to obtain confidence bounds on GE, since the T_j's are just functions of the sample and can be computed directly from the sample. However, we further simplify the inequality, using equation 4,

$$P[HE - GE \geq \frac{t}{m}] \leq e^{\frac{-2Nt^2}{m^3}}$$

We can assign $t' = \frac{t}{m}$, since m is a constant given the data generation model. With this we have,

$$P[HE - GE \geq t'] \leq e^{\frac{-2Nt'^2}{m}}$$

which is the desired inequality.

Proof We now present the proof for Lemma 2. Assume that we have r independent interactions of datapoints of size $m_1, m_2, ..., m_r$ such that $m_1 + m_2 + ... + m_r = N$ where N is the total number of datapoints. A distribution (i.e. data generation model) that models this scenario has the following form,

$$\begin{aligned} &P[X_{11}, Y_{11}, ..., X_{m_r r}, Y_{m_r r}] \\ &= P[X_{11}, Y_{11}, ..., X_{m_1 1}, Y_{m_1 1}] P[X_{12}, Y_{12}, ..., X_{m_2 2}, Y_{m_2 2}] \cdot ... \\ &\quad \cdot P[X_{1r}, Y_{1r}, ..., X_{m_r r}, Y_{m_r r}] \end{aligned}$$

where X_{ij} is the input space and Y_{ij} is the output space of the i^{th} datapoint in the j^{th} set of interactions ($j \in \{1, ..., r\}$ and $i \in \{1, ..., m_j\}$). A sample from this distribution produces N datapoints in the original space. The empirical error computed over these N datapoints is given as by,

$$HE = \frac{1}{N} \sum_{j=1}^{r} \sum_{i=1}^{m_j} \lambda(\zeta(x_{ij}, y_{ij}))$$

where x_{ij} and y_{ij} are values of X_{ij} and Y_{ij} respectively, in the sample.

As in the proof of Lemma 1 let $T_j = \sum_{i=1}^{m_j} \lambda(\zeta(x_{ij}, y_{ij}))$ where $j \in \{1, ..., r\}$. The empirical error is thus given by,

$$HE = \frac{1}{N} \sum_{j=1}^{r} T_j$$

Notice here though, that unlike in the case of Lemma 1 where the T_j's were i.i.d., the T_j's here are independent but not identically distributed. Hence, in this case we cannot directly apply the Hoeffding inequality as we did in the previous case.

We know that each T_j is the sum of m_j 0-1 loss functions, thus $T_j \in [0, m_j]$. We define new random variables Q_j such that $Q_j = k_j T_j$ where $k_j = \frac{l}{m_j}$ and l is the least common multiple of $m_1, m_2, ..., m_r$. Notice that since Q_j is a multiple of T_j and T_j can be computed from the sample, hence Q_j can also be computed from the sample. Moreover, though the Q_js are not identically distributed, they are independent as the T_js but additionally their mean is identical i.e. $E[Q_j]$ is the same $\forall j \in \{1, ..., r\}^3$. We define another random variable HE' (pseudo-empirical error) as follows,

$$HE' = \frac{1}{lr} \sum_{j=1}^{r} Q_j$$

We now derive the relationship between the expected value of $E[HE']$ and GE using linearity of expectation and the assumption that the marginals $P[X_{ij}, Y_{ij}]$ are equal $\forall i \in \{1, ..., m_j\}, j \in \{1, ..., r\}$.

$$\begin{aligned} E[HE'] &= \frac{1}{lr} \sum_{j=1}^{r} E[Q_j] \\ &= \frac{1}{l} E[Q_j] \\ &= \frac{k_j}{l} E[\sum_{i=1}^{m_j} \lambda(\zeta(x_{ij}), y_{ij})] \\ &= \frac{1}{m_j} \sum_{i=1}^{m_j} E[\lambda(\zeta(x_{ij}), y_{ij})] \\ &= \frac{1}{m_j} \sum_{i=1}^{m_j} GE \\ &= GE \end{aligned}$$

[3] The assumption of the marginals $P[X_{ij}, Y_{ij}]$ being equal $\forall i, j$ aids in the expected values being identical.

A subtle point regarding Hoeffding inequality is that the assumption of the random variables being i.i.d. to bound the average and their mean is an overkill. The inequality holds good even with the weaker assumption that the random variables are independent and have the same mean (other higher moments may vary). This can be seen by carefully examining the proof of Theorem 1 [16]. Thus, we can bound HE' and $E[HE']$ which is GE using the fact that $Q_j \in [0, l]$ $\forall j \in \{1, ..., r\}$ are independent and have the same mean. Thus,

$$P[\frac{1}{r}\sum_{j=1}^{r} Q_j - E[\frac{1}{r}\sum_{j=1}^{r} Q_j] \geq t] \leq e^{\frac{-2r^2t^2}{rl^2}}$$

we know that $HE' = \frac{1}{lr}\sum_{j=1}^{r} Q_j$ and that the $E[HE'] = GE$. With this we have,

$$P[HE' - GE \geq \frac{t}{l}] \leq e^{\frac{-2rt^2}{l^2}}$$

Here too we can assign $t' = \frac{t}{l}$, since l is a constant given the data generation model. We thus have,

$$P[HE' - GE \geq t'] \leq e^{-2rt'^2}$$

which is the desired result.

6.3 Derived inequalities and effective sample size

From Lemmas 1 and 2 we can see that the quality of the bound (i.e. its tightness) depends on the number of independent interactions rather than just the number of datapoints (N), which is the case in the i.i.d. setting. The reason for this is that such distribution free bounds are generally pessimistic i.e. the bound is derived assuming the worst possible behavior in the respective setting. Such bad behavior in our setting equates to having extremely high correlation between interacting datapoints. In the extreme case the correlation would be 1, which is equivalent to having just one datapoint in every set of interactions which makes the effective dataset size to be equal to the number of independent interactions. This notion of effective sample size was first introduced in [18], where the authors empirically observed the value of this quantity with varying amounts of auto-correlation (correlation between the same attribute in linked datapoints) and linkage (expected size of the interactions). They observed that with increasing linkage and auto-correlation the effective sample size reduced. A key aspect of the inequalities we derive is that, this notion of effective sample size directly pops out of our theory. In particular, the quantity $\frac{N}{m}$ or r (i.e. the number of independent interactions) serves as a conservative lower bound on the effective sample size.

7 Experiments

In the previous section we derived distribution free bounds for data generation models belonging to classes C1 and C2. We bounded the GE with HE (empirical error) for C1 models and the GE with HE' (pseudo-empirical error) for C2 models. Moreover, we related the derived inequalities to the notion of effective sample size. In this section we empirically evaluate the quality of these bounds on synthetic as well as real datasets. The experiments also show that a probability distribution over a given relational dataset which is consistent with the datasets connectivity can be modeled by the data generation models discussed in this paper.

7.1 Goals

1. **Application of the bounds in realistic settings:** We show that the data generation models considered in this paper are realistic and consequently the data generated by them can be used to train state-of-the-art relational classification algorithms. In addition, we show that the derived inequalities can be used to bound the GE of a classifier trained on real datasets.
2. **Intuitive appeal of the derived inequalities:** The experiments provide an intuitive feel of how the bounds behave in various settings, which points to circumstances when they are truly useful.

7.2 Factors affecting/not affecting the bound

Considering the above goals, it is pertinent that we clearly state what affects the tightness of the bound and what does not. This will aid us in designing experiments that help us attain our goals without unnecessarily complicating the respective experimental setups.

The following is a list of things that *do not affect* the quality of the bound but can be misconstrued to do so.

1. The number of attributes or dimensionality (d) of the space. This includes intrinsic as well as relational attributes.
2. The number of values of each attribute.
3. The particular classification algorithm.
4. The particular joint distribution. That is to say the specific probabilities of observing each input-output pair does not affect the bound.

The bound *only depends* on the underlying structure of the joint distribution which in turn depends on the test set or the *inference graph* [13]. An inference graph consists of objects and links just as a data graph. The only difference between the 2 graphs is that while a data graph represents the training set for a relational classification algorithm, the inference graph represents the test set. The structure of the inference graph provides information about the number of independent interactions which affects the tightness of the derived bounds.

7.3 Accomplishment of goals

To achieve the above stated goals we perform experiments on synthetic and real data. Since the bounds are only affected by the underlying structure of the joint distribution or the number of independent interactions in the test set, in the synthetic data experiments we generate training and test sets for each joint distribution whose underlying structure is either a particular fixed size interaction or a variable size interaction. We learn a relational classification algorithm on these generated training sets and estimate

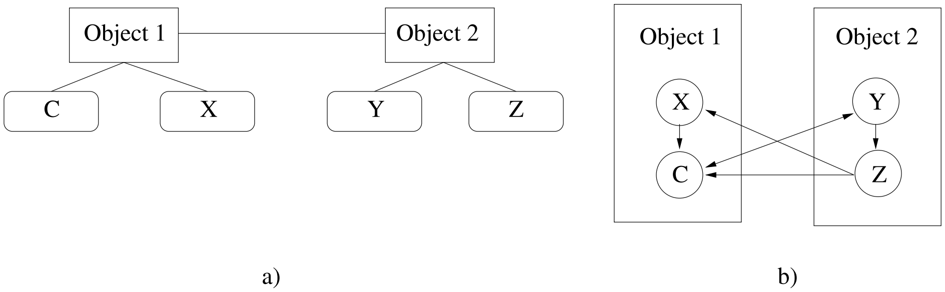

Fig. 6 a) represents a relational schema with types, *Object 1* and *Object 2*. The relationship between them is many-to-many. The rounded boxes denote their respective attributes. b) is the corresponding model graph which depicts the conditional dependencies between the attributes of the 2 types.

the error on the test set (or inference graph). 95% confidence bounds are then applied to this estimated error which gives us an idea of the GE. In the experiments on real data we choose a single real dataset, train the relational classification algorithm, estimate the error on inference graph and then apply the bound to this estimated error which depends on the structure of the inference graph. Though we perform experiments using a single relational classification algorithm and on a single real dataset they are a proof of concept that the data generation models considered are realistic and that the bounds are applicable in practical scenarios. Other relational classification algorithms [13] can be similarly trained on the data generated by these models and the bound similarly applied to other real datasets.

We now present in detail the relational classification algorithm used and the setup for the experiments on synthetic and real data.

Relational classification model: Relational Dependency Network

The relational classification model we use in the experiments is the Relational Dependency Network (RDN) [13]. The algorithm used to train this model and infer over it is given in [24]. We now briefly describe this model and the corresponding algorithm.

A RDN is a state-of-the-art PMRD that is known to perform well on relational data [27]. It (any PMRD in general) is characterized by three graphs namely; the data graph, the model graph and the inference graph. We have already seen that the data graph and the inference graph are made of objects and links and represent the training set and test set respectively. A model graph on the other hand represents the conditional dependencies between attributes of the same as well as related object types. An example model graph is seen in Figure 6b. The model graph in this figure, depicts the dependencies between 4 attributes C, X, Y and Z where C and X belong to type *Object 1* while Y and Z belong to type *Object 2*. The direction of the arrows in Figure 6b characterizes the nature of the dependence. In particular an arrow pointing to an attribute from another attribute implies that the first attribute is dependent on the other. Hence, in the figure X depends on Z while Z depends on Y. C and Y are both dependent on each other since there is a double headed arrow between them. Other dependencies in Figure 6b can be similarly deciphered. This dependency structure between attributes narrows down the conditional probability distributions (CPDs) that have to be learned by a RDN. For example since in Figure 6b X depends on Z, the

CPDs $P[X|Z = z]$ $\forall z \in Z$ have to be learned from the data. However, there is no need to learn $P[Z|X = x]$ $\forall x \in X$ since the corresponding dependency does not exist. Hence, in a RDN the conditional probability distributions that are to be learned are determined by the dependencies between attributes.

Learning: Now that we know the CPDs that have to be learned we use pseudo-likelihood techniques to learn them. Maximizing the log-likelihood of the overall joint distribution can prove to be expensive. Moreover, the CPDs considered do not factor the joint distribution. Hence, we maximize the pseudo-likelihood which is maximizing the log-likelihood of each individual CPD separately. Pseudo-likelihood estimators are unbiased estimators of the true values of the parameters [13]. The CPD estimation process is done using one of the two models namely; the Relational Bayes Classifier (RBC) [28] or the Relational Probability Tree (RPT) [27]. In the experiments below we use the RBC for CPD learning and hence we now describe only this relational learner. The RBC considers the values of the attribute on the left of the conditional sign (i.e. X in $P[X|Z]$) to have a multinomial distribution given the values of the attributes on the right (i.e. Z in $P[X|Z]$). The individual CPDs are estimated using these multinomials. The RBC does not perform feature selection and hence the attributes in the CPDs are defined by the model graph itself.

Inference: The RDN has to predict the class labels of the relevant objects of the same type over the inference graph. The inference graph can have multiple copies of the objects to be classified linked directly or through other objects (eg. papers linked through authors). To perform inference we first choose the learned CPDs that correspond to the attribute values in the inference graph. We thus have a bunch of CPDs with potentially multiple copies of some CPDs (since two different objects can have the same attribute values). These CPDs are then used to perform Gibb's sampling which provides samples from the joint distribution over the relevant attributes. The final prediction is done over the samples obtained from this joint distribution.

Synthetic data experiments

We now discuss the setup for the synthetic experiments. We explain below the data generation models or joint distributions used to generate data and bound the GE of the RDN classifier trained and tested on this generated data. In subsection 7.2 we mentioned which factors affect and which do not affect the bound. Considering those factors the following setup is sufficient to achieve our stated goals.

Joint distributions for data generation and bounding GE: A joint distribution over relational data is a distribution over the attributes of the related entities or objects. Hence, the first step in defining such a joint distribution is to define a relational schema with object types and relationships between these object types. The relational schema provides information about the attributes (single or multiple copies) that will comprise the joint distribution. A relational schema that we use in the synthetic experiments is given in Figure 6a. The schema has 2 types of objects namely; *Object 1* and *Object 2*. Each type has 2 attributes. *The attribute C of type* Object 1 *is the class attribute (i.e. the attribute whose value is to be predicted)* and the other attributes are explanatory attributes (i.e. X, Y and Z). Each attribute takes 2 values and the 2 types join on all values. In Figures 8a, 8b and 8c we consider data generated by joint distributions with fixed size interactions of size 2, 4 and 5 respectively. The joint distribution with fixed

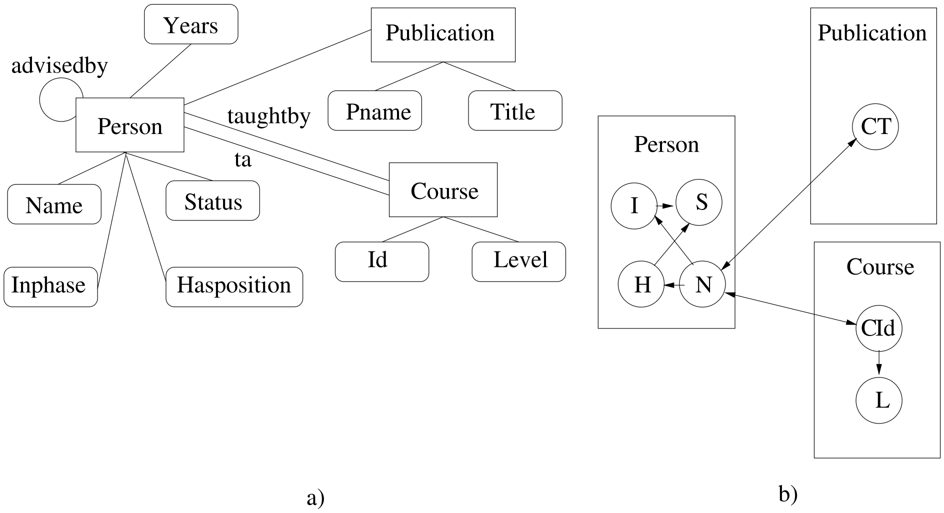

Fig. 7 a) represents a relational schema of a real dataset UW-CSE with types, *Person*, *Course* and *Publication*. The relationship between the related types is many-to-many. The rounded boxes denote their respective attributes. b) is the corresponding model graph which depicts the conditional dependencies between the chosen attributes of the 3 types namely; Name (N), Status (S), Inphase (I), Hasposition (H), Concatenated Titles (CT), Concatenated course Ids (CId) and Level (L).

size interactions of size s has the following form, $P[C^1, X^1, Y^1, Z^1, ..., C^s, X^s, Y^s, Z^s]$ where the superscripts $[1, ..., s]$ denote the corresponding copies of the respective attributes. Thus the parameter space of the joint distribution is 2^{4s} (since 4 attributes each having 2 values with s copies). We assign the probability for each assignment of attributes to be $\frac{1}{2^{4s}}$. This completely characterizes our data generation model for fixed size interactions. For variable size interactions seen in Figure 8d, we increase the number of independent interactions from 100 to 1000 to 10000. The case where we have 100 independent interactions the sizes of the individual interactions are: twenty interactions of size 10, fifty interactions of size 15 and thirty interactions of size 20. The sizes for 1000 independent interactions are: two hundred interactions of size 10, five hundred interactions of size 15 and three hundred interactions of size 20. The sizes for 10000 independent interactions are: two thousand interactions of size 10, five thousand interactions of size 15 and three thousand interactions of size 20. The probabilities for each assignment of values to the attributes in each of the independent interactions is set to $\frac{1}{2^{4s}}$ where s is the size of the corresponding interaction. With this, the data generation model for variable size interactions is also completely characterized.

RDN Learning and Inference: The model graph for the RDN is shown in Figure 6b. The corresponding conditionals are learned using using a RBC. The training set size is set to 1000 for distributions with fixed size interactions. Note that since the goal of these synthetic experiments is, 1) to show that an RDN (any PMRD in general) can be trained on the data generated by the joint distributions considered in this paper and 2) to observe the behavior of the bounds, as opposed to evaluating the RDN algorithm itself, the size of the training set is unimportant. On the other hand the size of the test set is important for scenarios where we have fixed size interactions since it affects

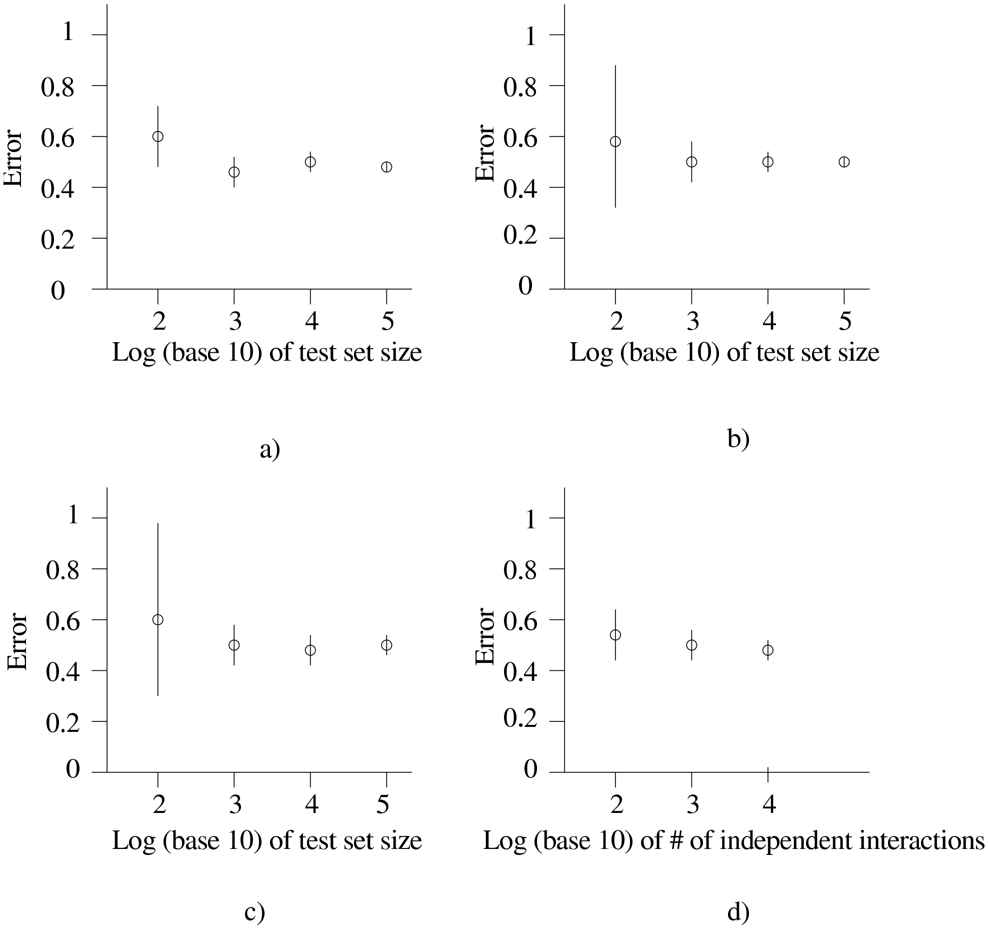

Fig. 8 95% confidence bounds for GE w.r.t. HE (a, b, c) and HE' (d) are shown for data generation models with a) interactions of size 2, b) interactions of size 4, c) interactions of size 5 and d) variable size interactions.

the tightness of the bound and hence we vary this parameter as observed in Figures 8a, 8b and 8c. For different test set sizes we use the required number of learned CPDs and perform inference using Gibbs sampling (burn-in 100, samples 1000). For variable size interactions we train and test on datasets of size 1550, 15500 and 155000 which have increasing number of independent interactions as seen in Figure 8d. Note that in this case HE' is computed and bounded with GE and not HE. Here too we provide the required number of copies of the learned CPDs and infer using Gibbs sampling (burn-in 100, samples 100).

Real data experiments

In experiments on real data we choose the UW-CSE dataset [21]. The UW-CSE dataset consists of people being either students or professors. The dataset has information regarding which course is taught by whom, who are the teaching assistants for a course, the publication record of a person, the phase in which a person is (i.e.

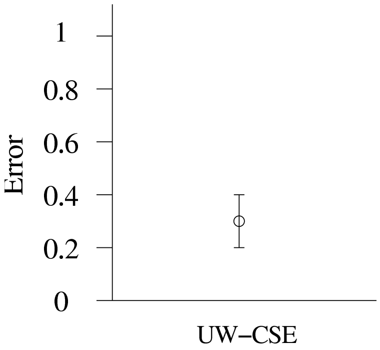

Fig. 9 95% confidence bound for GE w.r.t. HE' is shown on the UW-CSE dataset.

pre-qualifier, post-qualifier, post-general), the position of a person (i.e. faculty, affiliate faculty, adjunct faculty), years in a program and the advisor (or temporary advisor) of a student. A potential relational schema for this dataset is given in Figure 7a. The attribute Name refers to person name, *Status refers to the person being student or professor and is the class label in our experiments*, Inphase refers to the phase in which a student is, Hasposition refers to the position of a person, Years refers to the number of years a student has been in the program, Pname refers again to the person name, Title refers to the paper titles published by a person, Id refers to a specific course being offered and Level refers to the difficulty level of a course. The link advisedby relates students to their advisors (or temporary advisors), taughtby relates courses to their instructors and ta describes the respective teaching assistants for certain courses.

Having ellucidated the different components of the dataset we now explain how the bound is computed on the error made by a RDN on this dataset.

Joint distribution for bounding GE: We split the dataset into disjoint training and test sets with the training set size being 70% of the original dataset. The test set or the inference graph has variable size interactions and hence to compute the bound we count the number of independent interactions. The number of independent interactions turns out to be 57 which determines the width of the bound.

RDN Learning and Inference: The RDN is trained using the model graph given in Figure 7b. In the model graph we introduce 2 new attributes not present in the relational schema namely, CT and CId which are formed by concatenating the titles of papers written by a person and by concatenating Ids of courses taught (or ta) by a person. The Year attribute is eliminated since it is not particularly discriminative. The respective CPDs are learned using a RBC and the inference is performed using Gibbs sampling (burn-in 100, samples 1000) with each entity in the inference graph receiving its copy of the relevant CPD.

7.4 Observations and implications

In the Figures 8a, 8b, 8c we see that the width of the bound reduces rapidly with increasing test set size. Similarly, in Figure 8d the width of the bound reduces rapidly with increasing number of independent interactions. These synthetic experiments show that the data generation models considered in this paper generate data which is consistent with the structure of standard relational datasets and hence can be used to train relational classification algorithms developed by the SRL community. Moreover, they show that the derived bounds are tight enough for a reasonable choice of parameters (N, r). In Figure 9 we see that the bound is acceptable when applied to a real dataset which implies that the derived bound can be used in realistic settings. In addition, it strengthens our claim that the data generation models introduced by us are realistic and capture the complex dependencies in real relational datasets.

8 Discussion

In the previous sections we derived and evaluated bounds on the GE of a classifier in the presence of interactions. In this section we discuss ideas and lines of future research in an attempt to derive tighter bounds.

1. **Bounds w.r.t. different estimators of GE:** In Lemma 2 we bounded GE with HE' which is not the usual empirical error (HE) that occurs in the Hoeffding inequality applied to the classification problem in an i.i.d. setting. Since, our main concern is bounding GE, it really does not matter if we bound it w.r.t. HE or HE', as both these estimators can be computed from the sample. Thus, what we truly care about is being able to derive tight bounds w.r.t. some estimator of GE. This is precisely what we did in Lemma 2 where we bounded GE with HE'. In general, we need not limit ourselves in bounding GE with specific estimators such as HE used in literature, but rather should choose estimators for whom we can derive the *tightest* possible bounds and can be computed reasonably efficiently from the sample. One of the reasons GE was bounded w.r.t. HE in literature, was that the definition of HE allowed for direct application of the Hoeffding inequality and hence it was the most natural choice. But in the presence of interactions this may not be the case (as we have seen in Lemma 2) and we should choose the appropriate estimator w.r.t. which we can bound GE.
2. **Bounds w.r.t. different loss functions:** In the paper we derived bounds using the 0-1 loss function. However, nowhere in the proof of these bounds have we used any special properties that are specific to this particular loss function except that its value is between zero to one. Hence, the derived bounds can be directly applied to other loss functions whose values are between zero to one. An example of such a loss function that is commonly used in practice is the least squares loss for binary classification. For loss functions whose values do not lie between zero to one the bound can be easily adapted. For example, if $[a, b]$ is the interval in which the values of the loss function lie then the corresponding bound for C2 models (C1 models are just a special case of C2 models) is, $P[HE' - GE \geq t] \leq e^{\frac{-2rt^2}{(b-a)^2}}$. A common example of such a loss function is the exponential loss which widely used in machine learning.
3. **Bounds using algorithms:** Most of the distribution free bounds have simple closed form formulae. Such results are elegant, easy to use and can provide insight into the behavior of the random variables being bounded. However, from the point of view of obtaining tight bounds this can have an adverse effect since in the process of deriving simple formulae the bounds have to be invariably made loose. An example of this is the Hoeffding inequality itself where intermediate results

in the derivation of Theorem 1 are tighter [16], but are not as elegant. At the time when these bounds were derived, computers were not particularly widespread (if at all they existed) and consequently it made sense to derive simple formulae. However, in the present circumstances it is unnecessary to constrain ourselves to just simple closed form formulae but rather we can automate the procedure used in the deriving these bounds and without much loosening of the bound (bounds are generally made loose using convexity [19]) obtain much less elegant but tighter results. Moreover, in the presence of interactions, to obtain tighter bounds than the ones derived in this paper, it may not be possible to derive simple formulae in which case developing algorithms might be useful.

4. **Bounds as functions of correlation:** The bounds we derived in Lemmas 1 and 2 assumed the highest amount of correlation between interacting datapoints. A way of tightening the obtained bounds is to make the bound depend on the strength of correlation between the set of interacting datapoints. It is not very clear as to how this might be accomplished. A possible alternative would be to express the sample in terms of the information it possesses using information theoretic metrics such as entropy and then deriving the bounds using this statistic. If such bounds are derived they would most likely be tighter than the present ones. Moreover, they would provide us with better estimates of the effective sample size and not a lower bound. However, such bounds would need more information and may not be as simple to use or as intuitive as the ones derived in this paper.

9 Conclusion

In this paper we derived distribution free bounds for the GE of a classifier in the presence of dependencies. We related the derived bounds to the notion of effective sample size by explaining that the number of independent interactions is in fact a lower bound on this quantity. In the experiments we validated the claim that the data generation models considered in this paper are in fact realistic and that the bounds are tight enough to be applicable in practical scenarios. We also discussed strategies in obtaining tighter bounds which provides avenues to extend this type of work. In summary, we have taken an initial step towards finding useful distribution free bounds in the relational setting.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 0448264.

References

1. M. Arias, A. Feigelson, R. Khardon, and R. Servedio. Polynomial certificates for propositional classes. *Inf. Comput.*, 204(5):816–834, 2006.
2. M. Arias and R. Khardon. Learning closed horn expressions. *Inf. Comput.*, 178(1):214–240, 2002.
3. G. Bakir, T. Hofmann, B. Schölkopf, A. Smola, B. Taskar, and S. V. N. Vishwanathan. *Predicting Structured Data.* The MIT Press, 2007.

4. P. Bartlett, O. Bousquet, and S. Mendelson. Local rademacher complexities. In *Annals of Statistics*, pages 44–58, 2002.
5. G. Bennett. Probability inequalities for the sums of independent random variables. *JASA*, 57:33–45, 1962.
6. A. Blum, A. Kalai, and J. Langford. Beating the hold-out: Bounds for k-fold and progressive cross-validation. In *Computational Learing Theory*, pages 203–208, 1999.
7. A. Blumer, A. Ehrenfueucht, D. Haussler, and M. Warmuth. Occam's razor. *Information Processing Letters*, 24:377–380, 1987.
8. H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–507, 1952.
9. W. Cohen. Polynomial learnability and inductive logic programming: Methods and results. *New Generation Computing*, 13:369–409, 1995.
10. L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer-Verlag, 1996.
11. S. Floyd and M. Warmuth. Sample compression, learnability and the vapnik-chervonenkis dimension. In *Machine Learning*, pages 269–304, 1995.
12. N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *IJCAI*, pages 1300–1309, 1999.
13. L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning.* MIT Press, 2007.
14. H. Godwin. On generalization of tchebyshev's inequality. *JASA*, 50:923–945, 1955.
15. G. Grimmett and D. Stirzaker. *Probability and Random Processes.* Oxford, 3 edition, 2001.
16. W. Hoeffding. Probability inequalities for sums of bounded random variables. *JASA*, 58(301):13–30, 1963.
17. G. Hulten, P. Domingos, and Y. Abe. Mining massive relational databases, 2003.
18. D. Jensen and J. Neville. Linkage and autocorrelation cause feature selection bias in relational learning, 2002.
19. J. Jensen. Sur les fonctions convexes et les ingalits entre les valeurs moyennes. *Acta Mathematica*, 30:175–193, 1906.
20. Y. Jia, J. Zhang, and J. Huan. An efficient graph-mining method for complicated and noisy data with real-world applications. *Knowledge and Information Systems*, 2011.
21. S. Kok, P. Singla, M. Richardson, and P. Domingos. The alchemy system for statistical relational ai. Technical report, Department of Computer Science and Engineering, UW, http://www.cs.washington.edu/ai/alchemy/, 2005.
22. J. Langford. Tutorial on practical prediction theory for classification. *J. Mach. Learn. Res.*, 6:273–306, 2005.
23. D. Mcallester. Pac-bayesian model averaging. In *In Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, pages 164–170. ACM Press, 1999.
24. J. Neville. Statistical models and analysis techniques for learning in relational data. Ph.D. Thesis, University of Massachusetts Amhers, 2006.
25. J. Neville, B. Gallagher, T. Eliassi-Rad, and T. Wang. Correcting evaluation bias of relational classifiers with network cross validation. *Knowledge and Information Systems*, 2011.
26. J. Neville and D. Jensen. Leveraging relational autocorrelation with latent group models. In *MRDM '05: Proceedings of the 4th international workshop on Multi-relational mining*, pages 49–55, New York, NY, USA, 2005. ACM.
27. J. Neville and D. Jensen. Relational dependency networks. *J. Mach. Learn. Res.*, 8:653–692, 2007.
28. J. Neville, D. Jensen, and B. Gallagher. Simple estimators for relational bayesian classifiers, 2003.
29. M. Okamoto. Some inequalities relating to the partial sum of binomial probabilites. *Annals of the institute of Statistical Mathematics*, 10:29–35, 1958.
30. A. Papoulis. *Probability, Random Variables and Stochastic Processes.* McGraw-Hill, 3 edition, 1991.
31. C. Preisach and L. Schmidt-Thieme. Ensembles of relational classifiers. *Knowledge and Information Systems*, 14(2):249–272, 2008.
32. L. Raedt. First order jk-clausal theories are pac-learnable. *Artificial Intelligence*, 70:375–392, 1994.
33. C. Reddy and J. Park. Multi-resolution boosting for classification and regression problems. *Knowledge and Information Systems*, 2010.

34. M. Richardson and P. Domingos. Markov logic networks. *Mach. Learn.*, 62(1-2):107–136, 2006.
35. F. Rusu and A. Dobra. Pseudo-random number generation for sketch-based estimations. *ACM Trans. Database Syst.*, 32(2):11, 2007.
36. I. Savage. Probability inequalities of the tchebyshev type. *Journal of Research of the National Bureau of Standards*, 65B:211–222, 1961.
37. J. Schmidt, A. Siegel, and A. Srinivasan. Chernoff-hoeffding bounds for applications with limited independence. *SIAM J. Discrete Math*, 8:223–250, 1995.
38. B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *In Proc. 18th Conference on Uncertainty in AI*, pages 485–492, 2002.
39. V. Vapnik. *Statistical Learning Theory.* Wiley & Sons, 1998.

Authors Biographies

Fig. 10 **Amit Dhurandhar** is a research staff member in the Mathematical Sciences Dept. at IBM T.J. Watson. He received his B.E. in computer engineering from Pune University in 2004. He then received his Masters and P.h.d. in computer engineering from University of Florida in 2005 and 2009 respectively. Broadly speaking, Amit's research interests primarily span the areas of machine learning, data mining and computational neuroscience. He has authored several papers and has been a reviewer for many top quality conferences and journals.

Fig. 11 **Alin Dobra** is an associate professor at the University of Florida. He received his B.S. in computer science from Technical University in Cluj-Napoca in 1998. He then received his Masters and P.h.d. in computer science from Cornell University in 2001 and 2003 respectively. Alin's main interests are in approximate query processing and foundations of data mining. He has been a program committee member for several top conferences and a reviewer for top journals.