Appears in the
Proceedings of the Tenth Canadian Conference on Artificial Intelligence (CSCSI-94),
Banff, May 1994.

Learning Default Concepts

Dale Schuurmans
Department of Computer Science
University of Toronto, Toronto, ON M5S 1A4
dale@cs.toronto.edu

Russell Greiner
Siemens Corporate Research
Princeton, NJ 08540
greiner@learning.siemens.com

Abstract

Classical concepts, based on necessary and sufficient defining conditions, cannot classify logically insufficient object descriptions. Many reasoning systems avoid this limitation by using "default concepts" to classify incompletely described objects. This paper addresses the task of learning such *default concepts* from observational data. We first model the underlying performance task — classifying incomplete examples — as a probabilistic process that passes random test examples through a "blocker" that can hide object attributes from the classifier. We then address the task of learning accurate default concepts from random training examples. After surveying the learning techniques that have been proposed for this task in the machine learning and knowledge representation literatures, and investigating their relative merits, we present a more data-efficient learning technique, developed from well-known statistical principles. Finally, we extend Valiant's PAC-learning framework to this context and obtain a number of useful learnability results.

1 Introduction

Many reasoning tasks involve "classification" [Cla85] — *i.e.*, determining whether a particular object belongs to a specified class, given a description of that object. For example, a diagnosis process must determine whether a patient, with a specified set of symptoms, has a particular disease; a chess player must determine whether a particular move is appropriate given a board configuration; and a planner must determine whether to apply a particular action, given the perceived state. Many classifiers are based on *classical concept definitions* (ccds), which specify necessary and sufficient conditions for concept membership. While these systems can work effectively when given *completely specified* objects (*e.g.*, a complete description of the patient's symptoms, etc.), they may be unable to categorically classify objects that are only *partially* described.

Unfortunately, we may still have to provide a classification for such partially-described domain objects. For example, as doctors seldom have access to every potentially relevant fact about a patient, they usually cannot rule out all but the one true disease. The patient is usually better off if the doctor makes a credulous assessment and suggests some treatment based on what is known, rather than skeptically withholding judgement.

Notice that the doctor's diagnosis can *change* if he receives further information about the patient. As this type of *nonmonotonic* classification behavior cannot be described in terms of necessary and sufficient conditions, it cannot be encoded as a ccd. There are, however, formalisms designed to classify partial object descriptions. *Default concept definitions* (dcds) are a natural generalization of ccds, which avoid this limitation by using *default* classification rules [Rei87]. These classifiers play an important role in many expert systems [Cla85, PBH90].

Of course these dcds must somehow be acquired for such applications. As it is often quite difficult to explicitly extract the knowledge of domain experts, it makes sense to use machine learning techniques to automatically acquire the appropriate default concept based on existing "solved" cases; *cf.*, [PBH90]. Unfortunately, the task of learning default concept definitions has received relatively little attention, especially when compared to the vast literature on the subject of learning to classify *complete* object descriptions. To date, only a few empirical studies have been published [PBH90, Qui89, BFOS84], and the problem has yet to receive an adequate theoretical treatment in the machine learning literature; *cf.*, [Riv87, p.245]. This means there is no supporting theory that specifies when proposed techniques can be expected to perform well, or even why they work when they do.

We attempt to fill this void by studying the problem of learning accurate default concepts from examples within a precise mathematical framework. As preliminaries, Section 2 first defines the formal structure of default concepts and the associated object level classification task, and Section 3 introduces a probabilistic testing model that incorporates "attribute blocking". Section 4 then considers the problem of *learning* accurate dcds from random training examples: It considers learning under a relatively benign (resp., completely general) blocking model, introduces many of the exist-

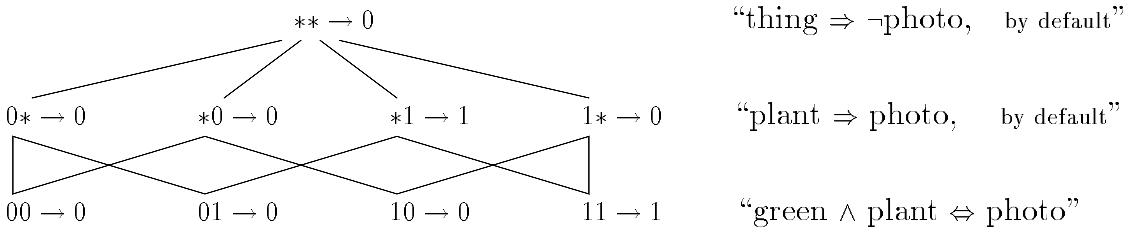

Figure 1: Structure of a complete default concept definition

ing learning techniques discussed in the literature, and considers an alternative procedure (relatively unknown in machine learning community) that is based on well-known ideas from theoretical statistics. It also extends Valiant's PAC-learning framework to the present case: assessing the effects of prior knowledge on learning efficiency, and determining the difficulty of learning under different conditions.[1]

We first close this introduction by tying this research to existing work: Notice first that, while there is a voluminous literature on default and nonmonotonic reasoning [Rei87], and even a recent trend towards probabilistic interpretations of default logics [Pea88, Bac90], the issue of *learning* defaults has scarcely been raised. Second, to avoid possible confusions, it is worth explicitly distinguishing our "missing attribute" framework from two other models of learning from the learnability community: A system that learns with attribute noise [SV88] does not know which attribute values have been corrupted; by contrast, we know explicitly which values are missing. Also, a probabilistic concept [KS90] is a mapping $c_i : X_n \mapsto [0,1]$ from the space of *complete* object descriptions X_n to probability values; such mappings do not directly handle missing attribute values.

2 Default Concepts

Following standard practice, we consider a set of domain objects $X_n = \{0,1\}^n$, where each object is identified by a vector of boolean attributes $\bar{x} = \langle x_1, \ldots, x_n \rangle$. A (complete) test example is specified by a pair $\langle \bar{x}, c \rangle$, consisting of a domain object $\bar{x}$ and its true classification c. In standard classification models, this domain object $\bar{x}$ would be passed "as is" to the classifier before testing its classification against the correct class c. Here, however we assume the classifier only sees a "degraded version" of $\bar{x}$ in which certain attribute values have been replaced by the "unknown" value $*$; see Figure 2. We model this degradation using a (stochastic) *blocking process* $\beta : X_n \times \{0,1\} \to \{0,1,*\}^n$ that may "hide" some of the attribute values: replacing certain values with $*$, but otherwise leaving $\bar{x}$ intact. Thus, $x_i = 0$ can be mapped to $x_i^* \in \{0,*\}$, and $x_i = 1$ to $x_i^* \in \{1,*\}$. We let $X_n^* = \{0,1,*\}^n$ denote the set of possible object *descriptions*. A *test example* $\langle \bar{x}^*, c \rangle$ is a (partial) description $\bar{x}^*$ of some domain object $\bar{x}$, along with $\bar{x}$'s true classification $c \in \{0,1\}$. The space of possible examples is denoted $X_n^* \times \{0,1\}$.

A *classical concept definition* (ccd) is a subset of X_n, which we represent by its indicator function $c : X_n \to \{0,1\}$; thus $c(\bar{x}) = 1$ iff $\bar{x}$ belongs to the concept. A *default concept definition* (dcd) $d : X_n^* \to \{0,1\}$, on the other hand, takes a *description* $\bar{x}^*$ as its input and returns $d(\bar{x}^*) = 1$ if the object described by $\bar{x}^*$ belongs to the concept *by default*, and returns 0 otherwise. Given a test example $\langle \bar{x}^*, c \rangle$, a dcd d makes a *correct* classification if $d(\bar{x}^*) = c$, otherwise it makes an *error*.

We can represent a dcd d as a collection of *rules* of the form $\bar{x}^* \to c$ where $c \in \{0,1\}$ and $\bar{x}^* \to c \in d$ means $d(\bar{x}^*) = c$. By insisting that for every object description $\bar{x}^* \in X_n^*$ either $\bar{x}^* \to 1 \in d$ or $\bar{x}^* \to 0 \in d$ but not both, we are in effect only considering *complete* dcds that categorically classify every possible object description, even $\bar{x}^* = \langle *, *, \ldots, * \rangle$.[2] To illustrate, consider the example of a dcd on two attributes shown in Figure 1, where the first attribute is "green", the second "plant", and the class is "photosynthetic".[3] Notice this collection of rules specifies *nonmonotonic* classification behavior, as its assessment of concept membership can change as more attributes are specified. For example, even though non-green–plants $\subset$ plants $\subset$ things, the predicted photosynthesis properties are 0, 1, 0, respectively. Such a classifier cannot be specified by a classical concept.[4]

There are many unexpected similarities between dcds and existing nonmonotonic knowledge representation formalisms. For example, Reiter [Rei87] considers commonsense concepts like "bird", "chair", and "game" and notes that they do not have classical definitions in terms of necessary and sufficient conditions. He argues that these concepts can be better characterized by specifying "default" necessary and sufficient conditions, and shows that this idea is similar to Minsky's concept of

[1] Unfortunately, space constraints preclude presenting proofs of the results stated in this abstract; see [Sch94].

[2] Thus there are 2^{3^n} distinct dcds possible on n boolean attributes. Only some of these have "reasonable" structures, see Lemma 1 below.

[3] Each node in the graph represents a rule; *e.g.*, "$*1 \to 1$" encodes the rule that plants, of unspecified color, are accepted in the photosynthetic class. An arc descending from node n_1 to n_2 means the antecedent of n_1's rule is "more general" than n_2's antecedent, in that any object that matches n_2's antecedent will also match n_1's.

[4] Notice the blocking process β introduces only a restricted form of ambiguity: β may produce descriptions corresponding to disjunctions like $0* \equiv 00 \vee 01$, but cannot produce a description corresponding to $01 \vee 10$ (this is reminiscent of [BE89]) — *i.e.*, it cannot express the claim that an object is "either a non-green plant or a green non-plant". This will restrict the type of "reference classes" we must consider when learning dcds; see Footnote 7 below.

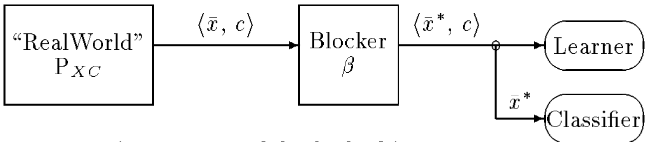

Figure 2: Model of Blocking Process

frames [Min75]: frame selectors can be viewed as "default" sufficient conditions for the frame concept, and frame instantiations can be viewed as "default" necessary conditions. These notions of non-classical concepts appear quite similar to the account of dcds developed here. Our acceptance conditions (rules of the form $\bar{x}^* \rightarrow 1$) correspond to Reiter's "default" sufficient conditions (a.k.a. frame selectors). However our rejection conditions (rules of the form $\bar{x}^* \rightarrow 0$) and Reiter's "default" necessary conditions (frame instantiations) are *contrapositives*, and do not serve precisely the same function [Gin87]. Still, the similarities are striking given the far different motivations behind these formalizations.

3 Model: Random Test Examples

We assume there is a "natural" source of random test examples against which we can evaluate the accuracy of any classifier. In particular, we assume there is a distribution P_{XC} over the space of domain objects and concept labels $X_n \times \{0,1\}$, called the *domain distribution*, from which random labelled objects are independently drawn. Before presentation, these labelled objects $\langle \bar{x}, c \rangle$ are first passed through a *blocking process* β to yield test examples $\langle \bar{x}^*, c \rangle$; see Figure 2. Thus, the domain distribution P_{XC} and the blocking process β induce a distribution P_{X^*C} over the space of possible examples, called the *example distribution*. The *accuracy* of a dcd d, written $P_{X^*C}(d)$, is defined as the probability that d correctly classifies a random test example. Note that in general a classifier's accuracy depends on both the domain distribution *and* the blocking process. We say that any example distribution P_{X^*C} for which d is optimal *satisfies* d.

Lemma 1 *For any example distribution P_{X^*C}, the optimally accurate dcd d includes the rule $\bar{x}^* \rightarrow c \in d$ whenever $P_{X^*C}\{\langle \bar{x}^*, c \rangle\} > P_{X^*C}\{\langle \bar{x}^*, \neg c \rangle\}$. Furthermore, for any dcd d, there is an example distribution P_{X^*C} which makes d non-trivially[5] optimal.*

We can therefore interpret any dcd d as asserting a collection of inequalities about the underlying example distribution. Notice the meaning of a rule $\bar{x}^* \rightarrow c$ depends not only on the (objective) distribution of domain objects in the world P_{XC}, but also on the (subjective) blocking process β, which specifies *how* information is received by the classifier. There are a number of reasonable assumptions one could make about β, but we restrict our attention to just two: *independent blocking* and *arbitrary blocking*.

[5] Here we are ruling out the "pure noise" case where $P_{X^*C}\{\langle \bar{x}^*, c \rangle\} = P_{X^*C}\{\langle \bar{x}^*, \neg c \rangle\} = 1/2$ for each $\bar{x}^* \in X_n^*$; here every dcd is (trivially) optimal.

3.1 Independent blocking

The *independent blocking* model, β_I, hides each object attribute x_i with a fixed probability p_i that is independent of x_i's value and those of the other attributes x_j, $j \neq i$. In this model, it turns out the optimally accurate dcd is determined strictly by the domain distribution P_{XC}, regardless of the specific blocking rates $\langle p_1, p_2, \ldots, p_n \rangle$.

Lemma 2 *Under β_I, for any domain distribution P_{XC}, the optimally accurate dcd d makes maximum conditional likelihood (mcl) classifications under P_{XC}, given the observed attributes of an object (cf.,* [DH73]*).*

Thus, the structure of an optimal dcd d is determined solely by the domain distribution, and we can interpret d as a collection of assertions about the domain distribution P_{XC} directly: $\bar{x}^* \rightarrow c \in d$ asserts that $P_{X^*C}\{\langle \bar{x}^*, c \rangle\} \geq P_{X^*C}\{\langle \bar{x}^*, \neg c \rangle\}$. However, not all of the possible 2^{3^n} dcds consistently specify mcl classifications in this manner — only (and all) the ones consistent with the following "consistent inheritance axiom."

Definition 1 (Consistent Inheritance) *A dcd d is* inheritance consistent *iff*

$$\left.\begin{array}{l} \langle x_1^* \ldots 0 \ldots x_n^* \rangle \rightarrow c \in d \\ \langle x_1^* \ldots 1 \ldots x_n^* \rangle \rightarrow c \in d \end{array}\right\} \Rightarrow \langle x_1^* \ldots * \ldots x_n^* \rangle \rightarrow c \in d.$$

Theorem 1 *Under β_I, d is inheritance consistent $\Longleftrightarrow$ d is satisfiable by some domain distribution P_{XC}.*

Existing default logics based on ϵ-semantics (*e.g.*, [Pea89]) all satisfy the consistent inheritance axiom and so tacitly assume independent blocking β_I. Here the meaning of a rule $\bar{x}^* \rightarrow c$ can be given a "majority" semantics under β_I akin to that of [Bac90].

3.2 Arbitrary blocking

While β_I is a simple and convenient model, it does not capture every practical situation; in particular, it cannot deal with circumstances where our knowledge of an attribute is *correlated* with its value; *e.g.*, ex-inmates are unlikely to answer the question "have you ever been in prison?". The *arbitrary blocking* model, β_A, can hide object attributes x_i according to an arbitrary probability distribution that can be conditioned on the entire object $\bar{x}$ and its classification c, allowing this model to incorporate correlations between hidden attributes and their values, other attributes, or even concept membership.

Under β_A the structure of an optimal dcd does not depend solely on the domain distribution P_{XC}, but also on the nature of the blocking process β. This means that making mcl classifications according to P_{XC} may no longer be optimal. In fact,

Lemma 3 *Under β_A, making mcl classifications according to P_{XC} can yield error rates arbitrarily close to $1/2$, even when the optimum dcd has error rate 0.*

Of course, other classifiers, which can exploit correlations between missing attributes and object classifications, can do much better in these situations.

4 Learning Accurate Default Concepts

We now consider the task of *learning* an accurate dcd from random training examples. We assume the learner L receives a sequence of random training examples drawn from a *training distribution*, from which it must produce a dcd, which is then tested on random test examples drawn from a *test distribution*. The learner's goal is to produce an accurate dcd with as few training examples as possible. We can consider a number of distinct learning problems, based on our assumptions about the form of training examples and the type of blocking process. Here, we focus the two types of blocking introduced in Section 3, and on the following two types of training examples.

The *incomplete* training example model, χ_I, assumes training examples are generated by the *same* example distribution that generates test examples. This is a natural model for many practical settings where we do not have access to complete object descriptions, even for training examples. One benefit of training on partial examples is that learner is exposed to the natural blocking process operating in the domain.

The *complete* training example model χ_C, on the other hand, assumes training examples are generated by the same *domain* distribution P_{XC} underlying the process that generates incomplete test examples. Here, however, some teacher has "filled in" the proper value of each attribute of each training example. Even though our goal is to learn classification rules that classify incomplete examples, we can still consider learning from *complete* examples. This situation that can easily arise in practical situations; *e.g.*, a medical student may be trained to diagnose the presence of a particular disease given fairly complete descriptions of all relevant patient data, and yet as a doctor, be expected to produce diagnoses without the benefit (and cost) of performing every available diagnostic test. Furthermore, we intuitively expect an advantage in training on complete examples as they appear to provide more information than incomplete examples. We will see below that this intuition is only sometimes correct.

4.1 Learning under Independent-Blocking

We first consider learning under the independent blocking model β_I. This is the simplest and arguably most natural blocking model, where the fact that an attribute is missing provides no information about the underlying values or object classifications. Lemma 2 showed that under β_I, the structure of the optimal d_{opt} depends solely on the domain distribution P_{XC}, regardless of the blocking probabilities $\langle p_1, \ldots, p_n\rangle$. In particular, d_{opt}'s classifications depend on the most probable class (under P_{XC}) given the *observed* (non-$*$) attributes of a description $\bar{x}^*$; *i.e.*, if $\mathrm{P}_{C|X}(c \mid obs(\bar{x}^*)) > \mathrm{P}_{C|X}(\neg c \mid obs(\bar{x}^*))$ then $\bar{x}^* \rightarrow c \in d_{opt}$. Hence, under β_I, learning an accurate dcd requires only determining whether $\mathrm{P}_{C|X}(c = 1 \mid obs(\bar{x}^*)) > \mathrm{P}_{C|X}(c = 0 \mid obs(\bar{x}^*))$ for each object description $\bar{x}^*$, based on observing a sequence of training examples.

4.1.1 Estimating Most Likely Classifications

Complete training examples: Here the learner is given a sequence of random training examples $\langle\langle\bar{x}_1, c_1\rangle, \ldots, \langle\bar{x}_m, c_m\rangle\rangle$ (drawn independently from the domain distribution P_{XC} — the same domain distribution that will be used to generate pre-blocked test examples), from which it must decide whether to use the classification rule $\bar{x}^* \rightarrow 1$ or $\bar{x}^* \rightarrow 0$ for each description $\bar{x}^*$. Here, it seems reasonably obvious that this decision should be based on the *observed* classification frequencies among all training examples $\bar{x}$ that *match* a description $\bar{x}^*$, as specified by the following learning strategy.

MLC (Maximum Likelihood (Complete)) For description $\bar{x}^*$, predict the most frequent class among all training examples whose domain object *matches* $\bar{x}^*$.

This simple strategy turns out to have the following rather remarkable optimality property.

Theorem 2 *For any learner L[6] that produces the* optimal *rule $\bar{x}^* \rightarrow c$ for some $\bar{x}^*$ with higher probability than* MLC, *given some P_{XC} and sample size m, there is another domain distribution P'_{XC} for which L produces a dcd d with accuracy $< 1/2$ with probability $> 1/2$.*

Thus, no learner can outperform MLC on *any* non–pure-noise domain distribution (*i.e.*, where $\mathrm{P}_{X^*C}\{\langle\bar{x}^*, c\rangle\} \neq 1/2$ for some $\bar{x}^*$), and object description.

Incomplete training examples: Here the learner is given a sequence of random training examples $\langle\langle\bar{x}^*_1, c_1\rangle, \ldots, \langle\bar{x}^*_m, c_m\rangle\rangle$ (drawn independently from the same example distribution P_{X^*C} used to generate test examples), from which it must decide whether to use classification rule $\bar{x}^* \rightarrow 1$ or $\bar{x}^* \rightarrow 0$ for description $\bar{x}^*$. As before, the optimal classification rules are determined by the underlying domain distribution P_{XC}, and so the general idea is to gain as much information as possible about P_{XC} from the random training examples; the difficulty here is that many of the training object attributes will be blocked. The challenge, therefore, is to extract as much information as possible from the object attributes that are actually observed.

A number of techniques have been proposed in the machine learning literature for determining the most likely classification of a description from a collection of incomplete training examples. Surprisingly, none of these techniques appear to make the most efficient use of the available training data. This leads us to investigate a simple statistical principle, relatively unused in machine learning, that appears to be far more efficient for this purpose. We first briefly survey the existing proposals and point out the intuitive source of inefficiency in each.

The first technique ignores the fact that training descriptions are independently blocked versions of complete descriptions, and simply gathers separate statistics for each description $\bar{x}^*$; effectively treating "$*$" as a third attribute value.

[6] Given the benign assumption that L's guesses for a description $\bar{x}^*$ are conditionally independent of the training labels of domain objects $\bar{x}$ that do *not* match $\bar{x}^*$.

THV (Three-valued) [Qui89] For description $\bar{x}^*$, predict the most frequent classification among training examples of the form $\langle \bar{x}^*, c \rangle$.

THV clearly does not make the most effective use of the available training data, given that attributes are blocked independently of their values. In particular, it ignores more specific training patterns that might match the description $\bar{x}^*$, which is ineffective as these patterns can provide additional information about the prevalence of a particular classification among objects matching $\bar{x}^*$. The next refinement is a technique that takes just this information into account.

LEM (Local error minimization) For description $\bar{x}^*$, predict the most frequent class among *all* more specific training patterns that *match* $\bar{x}^*$.

By considering more *specific* training patterns, LEM makes more efficient use of the training data than THV. However, it turns out that even LEM does not fully exploit all of the relevant information that can be gleaned from the training examples. In fact, there are situations where we ought to incorporate statistics from *more general* descriptions than $\bar{x}^*$. To illustrate this, imagine a simple setting where domain objects are described by a single bit, so any dcd for this domain will consist of three rules: $\{ \langle 0 \rangle \to c_0,\ \langle 1 \rangle \to c_1,\ \langle * \rangle \to c_* \}$ where each each $c_i \in \{0, 1\}$. Now, imagine a collection of training examples where

$$\begin{array}{lll} \#\langle\langle 0\rangle, 0\rangle = 2 & \#\langle\langle 1\rangle, 0\rangle = 2 & \#\langle\langle *\rangle, 0\rangle = 0 \\ \#\langle\langle 0\rangle, 1\rangle = 1 & \#\langle\langle 1\rangle, 1\rangle = 1 & \#\langle\langle *\rangle, 1\rangle = 14. \end{array}$$

Here, since $\#\langle\langle 0\rangle, 0\rangle > \#\langle\langle 0\rangle, 1\rangle$, it appears $\langle 0\rangle \to 0$ would be the optimal rule for $\langle 0\rangle$; similarly $\#\langle\langle 1\rangle, 0\rangle > \#\langle\langle 1\rangle, 1\rangle$ suggests $\langle 1\rangle \to 0$. Notice, however, that all 14 of the $\langle * \rangle$ observations belong to class 1, and *each of these must have actually been a domain object with attribute value* $\langle 0\rangle$ *or* $\langle 1\rangle$. So there is overwhelming evidence that at least one of the two attribute values (if not both) should be classified 1 rather than 0. This is a clear case where the statistics from a more general description should override those of the more specific.

A learning technique that attempts to do just this has been proposed in the philosophy of statistics literature — namely Kyburg's proposals for choosing the best reference class on which to base statistical judgements.

REF (Reference class) [Kyb83, Kyb91] For description $\bar{x}^*$, first select a "reference-class" description $\bar{x}_r^*$ (either $\bar{x}^*$ itself, or possibly a more general description), then predict the most likely classification given all training descriptions that *match* the reference class description $\bar{x}_r^*$.

The idea is to select a sufficiently general description $\bar{x}_r^*$ so that our choice of classification rule $\bar{x}^* \to c$ for $\bar{x}^*$ is based on "adequate" statistics. Kyburg suggests the following reference class selection procedure: For each incomplete description $\bar{x}^*$, compute a 90% (say) confidence interval about the probability of observing classification c given all training descriptions that match $\bar{x}^*$. Then employ a conflict resolution strategy (which trades-off interval bias and width) to decide whether to adopt, for this $\bar{x}^*$, the classification associated with successively more general reference classes [Kyb91].[7]

Although the REF strategy can override the predictions from specific descriptions with those from more general descriptions, it is not clear that it does so in the best conceivable way. The strategy is fundamentally *ad hoc* (in particular by incorporating an arbitrary parameter in the confidence intervals), and is not based on any real principles beyond "intuition" to adjudicate between candidate reference class descriptions. Furthermore, there is no empirical data to support the efficacy of this approach.

It is often stated that the crux of this type of statistical reasoning is the problem of "choosing the right reference class" [Bac90, BGHK92]. However, this premise might actually be leading us away from the most effective learning approaches here. Fundamentally, our goal should be to preserve *all* available statistical information, rather than throwing away statistics from one class in favor of those from another. The best approach should involve *combining* all of the available statistics in a principled way. Here we note that a well-known idea from theoretical statistics is applicable: namely, first determine the *maximum likelihood* distribution that accounts for *all* the data, then perform inferences according to this distribution [LR87]. This approach yields an effective method for determining the most likely classifications given incomplete training examples.

MLI (Maximum Likelihood (Incomplete)) [LR87] First, determine the domain distribution P_{XC}^{max} that maximizes the likelihood of the observed training examples. Then, for description $\bar{x}^*$, predict the most probable classification according to P_{XC}^{max}, given $\bar{x}^*$'s *observed* attributes.

Notice that this approach never "throws away" an observation; instead, it seeks the best model that accounts for all of them. The statistics for *all* relevant descriptions, both more general and more specific than $\bar{x}^*$, are combined in a principled way to yield a classification.

Based on the preceding discussion it seems intuitive that MLI should be more efficient than the other learning strategies, *i.e.*, we expect that MLI should produce more accurate classification rules, given fewer training examples. Although an optimality result akin to Theorem 2 has not yet been proven, it is fairly easy to demonstrate the superior efficiency of MLI empirically.

To support this point, consider the results of the following simulation study: Each of the four techniques was implemented and tested in the simple domain where domain objects are described by a single bit (as before). We then tested the techniques on random domain distributions and blocking rates, and recorded the accuracies

[7] Philosophical discussions often mention the difficulty in choosing the candidate reference classes to participate in any conflict resolution procedure (*cf.*, [Bac90, Chapter 5]). Kyburg simply adopts the reference classes considered here, and ignores other "disjunctive" classes (*cf.*, Section 2) by fiat. However, there is a principled argument behind ignoring disjunctive classes, based on the observation that they do not correspond to any possible "partial states of knowledge" one can have about a domain object, *cf.*, Section 2.

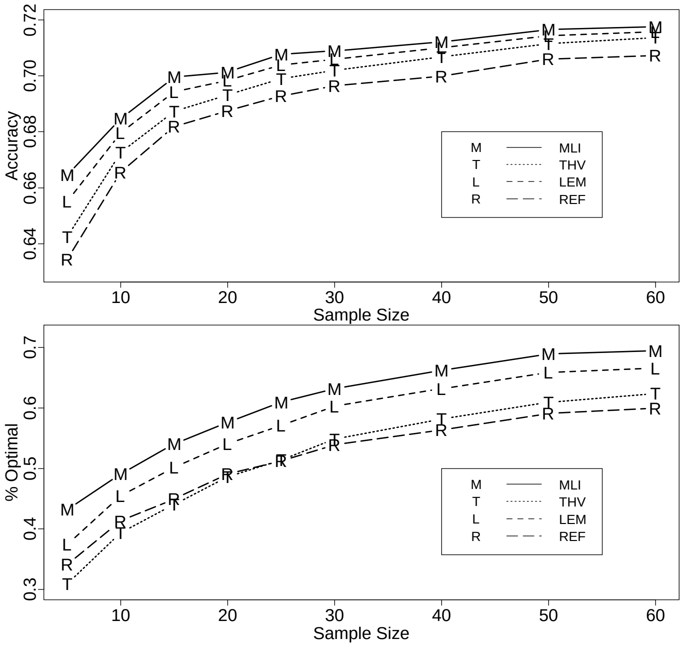

Figure 3: Percent Optimal, Accuracy vs Training Size

of the classification rules produced by each strategy. The graphs in Figure 3 plot the average accuracy obtained by each learner (resp., how often each learner returned the optimal dcd), as a function of training sample size; averaged over 10,000 trials. It is clear that, for a given number of training examples, MLI both attains the highest average accuracy levels, and also identifies the optimal dcd with the highest probability, *cf.*, Theorem 2.

4.1.2 Scaling Up

As efficient as the previous estimation techniques appear to be (particularly MLC and MLI), they cannot be applied "as is" to any real learning task. The problem, of course, is that these estimation techniques simply do not *scale up*. This is because determining the appropriate classifications for arbitrary object descriptions $\bar{x}^*$ can, in general, involve the simultaneous estimation of an exponential number of parameters (in n). For example, there are $\binom{n}{\lceil n/2 \rceil}$ descriptions containing $\lceil n/2 \rceil$ $*$'s, and most of the observations for one such pattern does not match any of the others. None of the estimation techniques generalize between these patterns.

This is a well-known issue in machine learning: to achieve reasonable performance with reasonable amounts of data, we will eventually have to introduce some form of prior knowledge to constrain our learning systems. This points to the necessity of *bias*. In any successful application, the learning system must be constrained to search a restricted space of appropriate classifiers, which here are dcds.[8]

Following the methodology pioneered by Valiant [Val84], we consider how learning performance *scales* as a function of prior knowledge. Here we *quantify* bias by its measurable effects on the quality of learning that can be guaranteed. *A la* Valiant, we consider prior domain knowledge that can be expressed by a restricted *set* of dcds $\mathcal{D}$, which is known to include the *optimal* dcd. The difficulty of learning a set of dcds $\mathcal{D}$ is then measured by the number of training examples needed to reliably guarantee a near optimal hypothesis, in the worst case over all possible example distributions satisfying some dcd $d \in \mathcal{D}$.

Definition 2 (PACO-learning) [9] *A learner L PACO-learns a class of dcds $\mathcal{D}$ under β_I blocking given $m(\epsilon, \delta)$ χ-type training examples ($\chi \in \{\chi_C, \chi_I\}$), if $\forall \epsilon > 0, \forall \delta > 0$, and $\forall$ domain distributions P_{XC} consistent with some $d_{opt} \in \mathcal{D}$, L outputs a dcd $d_L \in \mathcal{D}$ whose accuracy is within ϵ of this d_{opt}, with probability at least $1 - \delta$.*

To investigate scaling, we consider parameterized classes $\mathcal{D}_n$ defined on n attributes for $n = 1, 2, \ldots$.

Definition 3 (Feasible-learnability) *A parameterized class of dcds $\mathcal{D}_n$, $n = 1, 2, \ldots$ is said to be* feasibly-learnable *if there exists a polynomial function $poly(\cdots)$ and a learner L that PACO-learns each $\mathcal{D}_1, \mathcal{D}_2, \ldots$ with sample size $m(\epsilon, \delta) = poly(\frac{1}{\epsilon}, \frac{1}{\delta}, n)$.*

Intuitively, we expect the difficulty of learning a set of dcds $\mathcal{D}$ to depend on the "complexity" of $\mathcal{D}$, *i.e.*, more complex $\mathcal{D}$s are harder to learn. The question is: what precise complexity measure (effectively measuring the "amount" of prior knowledge encoded by $\mathcal{D}$) actually determines the difficulty of PACO-learning a default concept class $\mathcal{D}$? It turns out the appropriate complexity measures can be based on the notion of the *Vapnik-Chervonenkis dimension* of a set of dcds $\mathcal{D}$, written VCdim($\mathcal{D}$).[10]

Learning performance also clearly depends on the precise learning model under consideration (*e.g.*, β_I blocking and either χ_C or χ_I training examples). For β_I blocking and complete training examples, we have been able to identify precise conditions on the complexity of $\mathcal{D}_n$ (as a function of n) that determine whether $\mathcal{D}_n$ is feasibly learnable.

Lemma 4 *Under β_I blocking, $\mathcal{D}_n$ is feasibly-learnable from complete training examples $\iff \forall s \subset \{1, \ldots, n\}$,* $\mathrm{VCdim}(\mathcal{D}_n^s) = poly(n)$ *(where $\mathcal{D}_n^s$ is the set of ccds induced by $\mathcal{D}_n$ on attribute subset s).*

In the case of learning from incomplete training examples, a much stronger condition can be shown to be sufficient for feasible-learning.

Lemma 5 *Under β_I blocking,* $\mathrm{VCdim}(\mathcal{D}_n) = poly(n)$ $\Longrightarrow$ *$\mathcal{D}_n$ is feasibly-learnable from incomplete examples.*

[8] THV and MLI are particularly well suited to incorporating background knowledge; as demonstrated for THV in many decision-tree applications [Qui89, BFOS84], and for MLI by applications of the EM algorithm to parameterized domain distributions [LR87].

[9] For "Probably Approximately Class Optimal". Our goal differs slightly from standard PAC-learning, as we are forced to seek near-*optimal* rather than near-*perfect* classifiers, since with blocking *no* classifier can attain perfect accuracy in general. Notice also that we are only addressing the *sample* complexity of learning, *not* computational complexity.

[10] This is the same measure used when learning ccds. See [BEHW89] for a precise definition of VCdim and its application to determining the difficulty of learning sets of ccds.

Combining these lemmas yields the intuitive result that learning from complete training examples is easier than learning from incomplete examples:

Corollary 1 *$\mathcal{D}_n$ is feasibly-learnable under (β_I, χ_I) $\Longrightarrow \mathcal{D}_n$ is feasibly-learnable under (β_I, χ_C).*

However, the converse (*i.e.*, is there a class $\mathcal{D}_n$ that is is feasibly-learnable from complete but not incomplete training examples) remains an open question.

4.2 Learning under Arbitrary-Blocking

We now consider the arbitrary blocking model β_A. In this model, the fact that an attribute is missing from an object description can be correlated in various ways with the attribute values and the object's classification. In effect, *no* reliable information can be obtained about the value of missing attributes under β_A. Here, learning an accurate dcd amounts to determining whether $P_{C|X^*}(c = 1 \mid \bar{x}^*) > P_{C|X^*}(c = 0 \mid \bar{x}^*)$ for each object description $\bar{x}^*$, given training examples.

4.2.1 Estimating Most Likely Classifications

As in Section 4.1.1, we can consider the problem of estimating the most likely classification of a description $\bar{x}^*$ from both complete and incomplete training examples. The relative merits of the various learning techniques discussed in Section 4.1.1 change dramatically under these alternative learning conditions.

Complete training examples: Notice that *complete* training examples provide no information about the blocking process that will be applied to future test examples. By observing complete training examples, the learner can only estimate properties of the *domain* distribution P_{XC}, and not the *test example* distribution P_{X^*C} (generated by a blocking process over P_{XC}). Therefore it is fundamentally impossible to estimate whether $P_{C|X^*}(c = 1 \mid \bar{x}^*) > P_{C|X^*}(c = 0 \mid \bar{x}^*)$ for arbitrary blocking processes just by observing complete training examples. Lemma 3 exploits this fact to show that even given *exact* knowledge of the domain distribution P_{XC}, any classification rule produced by a learner can still have an arbitrarily high error rate on incomplete test examples for some example distribution P_{X^*C}. Therefore *no* learning strategy can reliably estimate the proper classification of an incomplete test description $\bar{x}^*$ from complete training examples.

Incomplete training examples: Given *incomplete* training examples, however, the learner is directly exposed to the natural blocking processes operating in the domain. Under these conditions it is possible to estimate whether $P_{C|X^*}(c = 1 \mid \bar{x}^*) > P_{C|X^*}(c = 0 \mid \bar{x}^*)$ for a description $\bar{x}^*$, simply by applying the THV strategy of determining whether $\#\langle \bar{x}^*, 1\rangle > \#\langle \bar{x}^*, 0\rangle$.

The various learning techniques discussed in Subsection 4.1.1 have different relative merits under the different learning conditions: We saw in Subsection 4.1.1 that LEM and MLI were more efficient than THV under β_I blocking. In general, maximum likelihood estimation (MLC, MLI) appears to be the superior technique for estimating the most probable classifications under β_I, regardless of whether complete or incomplete training examples are available. However, since these techniques base their judgements directly on estimated properties of the *domain* distribution P_{XC}, Lemma 3 shows that their classifications can have arbitrarily high error rates under β_A. In contrast, THV is the *only* provably effective technique for learning under β_A, given incomplete training examples, and so clearly dominates in this case.

These theoretical observations can actually help explain some of the results obtained by recent empirical studies: Quinlan [Qui89] compared applications of the LEM and THV techniques (along with some other ad hoc approaches) to decision-tree learning, and found that no single technique dominated the others over the set of test problem he considered. The preceding theoretical results, however, clearly demonstrate that the relative effectiveness of particular learning strategies strongly depends on the nature of the *blocking* process involved; an observation that can be applied in practice. For example, if blocking is known to be (more or less) independent (β_I), then MLI should outperform the other techniques, however, if blocking were known to be strongly correlated (β_A), then THV should dominate.

4.2.2 Scaling Up

As in Subsection 4.1.2, we can determine what constraints on prior knowledge (expressed as a parameterized class of dcds $\mathcal{D}_n$) are sufficient to permit efficient learning, as we scale up in n.

Lemma 6 *Under β_A, $\mathcal{D}_n$ is feasibly-learnable from incomplete examples $\Longleftrightarrow$ $\mathrm{VCdim}(\mathcal{D}_n) = poly(n)$.*

Notice that although complete training examples actually make learning *easier* under β_I, they make learning *impossible* under β_A. This is because complete examples provide information only about instance distribution, but supply *no* information about the blocking process that will be applied to future test examples. While this is not a problem under β_I (where the optimal classifications are determined strictly by the instance distribution P_{XC}), this issue is fatal under β_A; *cf.*, Lemma 3.

Lemma 7 No *non-trivial set $\mathcal{D}$ of default concepts is* PACO*-learnable under (β_A, χ_C).*

As expected, the feasible learnability of a parameterized class of dcds $\mathcal{D}_n$ depends on the specific conditions in which learning takes place. Here we compare the relative difficulty of learning under the various conditions.

Lemma 8 *$\mathcal{D}_n$ is feasible-learnability under (β_A, χ_I)*
$\Longrightarrow \mathcal{D}_n$ is feasible-learnability under (β_I, χ_I)
$\Longrightarrow \mathcal{D}_n$ is feasible-learnability under (β_I, χ_C).

The first inclusion is strict, as

Lemma 9 *There are parameterized classes $\mathcal{D}_n$ which are feasibly-learnable under (β_I, χ_I), but not feasibly-learnable under (β_A, χ_I).*

Hence, learning under (β_I, χ_I) is fundamentally easier than learning under (β_A, χ_I), as it can require exponentially fewer training examples in some cases.

5 Conclusions

This work constitutes a start on the general problem of acquiring default knowledge from empirical observations. Of course, much remains to be done. One of the more immediate concerns is to develop an efficient implementation of the MLI strategy for useful forms of bias. We are also beginning to examine many extensions to better cope with practical problems. For example, many application domains like medical diagnosis have the property that missing attribute values actually give *useful* information — namely that the missing attributes are *irrelevant* to the classification, given the known attribute values [PBH90]. Notice that β_I is overly restrictive and β_A is too underconstrained to adequately model such tasks; [GHR94] provides an initial analysis of this situation. We are currently investigating other intermediate blocking models that can more accurately model such domains and (we hope) lead to better empirical learning performance.

Other interesting research directions involve alternative generalizations of standard classification learning: This work has assumed that default definitions categorically classify every description, no matter how incomplete. An interesting direction is to consider *partial* default definitions that sometimes say "I don't know" *à la* [RS88]. Such classifiers could prove useful in domains where the consequences of an incorrect classification sometimes outweigh those of remaining silent.

Another interesting extension is to consider *active* classifiers. That is, we have assumed that classifiers *passively* observe test examples and play no role in determining which attributes are observed. It would be interesting to consider learning classifiers that *actively* decide which attributes to test, and when there is sufficient information to posit an accurate prediction (*i.e.,* learning to *diagnose*). This raises the issue of how best to trade off the number of tests required against the accuracy of the classifier.

Contributions: We formulated and studied the problem of learning "default concepts" (dcds), which can then be used to classify incomplete object descriptions. After formally defining the structure and function of dcds, we modelled the classification (performance) task as a random example generator that passes examples through a "blocking process" that hides object attributes from the classifier. We then addressed the task of learning dcds from random examples — first discussing many of the standard techniques for this problem, and then explaining why MLI is more effective than many standard learning techniques under the (β_I, χ_I) model. We also extended Valiant's PAC-learning framework to the problem of learning dcds: assessing the effects of prior knowledge on learning efficiency, and determining the difficulty of learning under different conditions. By providing a theoretical understanding of many empirical observations in the literature, we hope that our results will lead to the development of more effective learning procedures for practical problems that involve missing data.

References

[Bac90] F. Bacchus. *Representing and Reasoning with Probabilistic Knowledge.* MIT Press, 1990.

[BE89] A. Borgida and D. Etherington. Hierarchical knowledge bases and efficient disjunctive reasoning. In *KR-89*, 1989.

[BEHW89] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. of ACM*, 36(4), 1989.

[BFOS84] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth, Belmont, CA, 1984.

[BGHK92] F. Bacchus, A. Grove, J. Halpern, and D. Koller. From statistics to beliefs. In *AAAI-92*, 1992.

[Cla85] W. Clancey. Heuristic classification. *Artificial Intelligence*, 27, 1985.

[DH73] R. O. Duda and P. Hart. *Pattern Classification and Scene Analysis.* Wiley, 1973.

[GHR94] R. Greiner, T. Hancock, and R. B. Rao. Knowing what doesn't matter. Technical report, Siemens Corporate Research, 1994.

[Gin87] M. Ginsberg, editor. *Readings in Nonmonotonic Reasoning.* Morgan Kaufmann, Los Altos, 1987.

[KS90] M. J. Kearns and R. E. Shapire. Efficient distribution-free learning of probabilistic concepts. In *FOCS-90*, 1990.

[Kyb83] H. Kyburg. The reference class. *Philosophy of Science*, 50, 1983.

[Kyb91] H. Kyburg. Evidential probability. In *IJCAI-91*, 1991.

[LR87] J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data.* Wiley, 1987.

[Min75] M. Minsky. A framework for representing knowledge. In *The Psychology of Computer Vision.* McGraw-Hill, 1975.

[PBH90] B. Porter, R. Bareiss, and R. Holte. Concept learning and heuristic classification in weak-theory domains. *Artificial Intelligence*, 45, 1990.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, 1988.

[Pea89] J. Pearl. Probabilistic semantics for nonmonotonic reasoning: A survey. In *KR-89*, 1989.

[Qui89] J. R. Quinlan. Unknown attribute values in induction. In *ML-89*, 1989.

[Rei87] R. Reiter. Nonmonotonic reasoning. *Annual Review of Computer Science*, 1987.

[Riv87] R. Rivest. Learning decision lists. *Machine Learning*, 2(3), 1987.

[RS88] R. Rivest and R. Sloan. Learning complicated concepts reliably and usefully. In *AAAI-88*, 1988.

[Sch94] D. Schuurmans. *Efficient, Accurate, and Reliable Machine Learning.* PhD thesis, Univ. of Toronto, Dept. Computer Science (forthcoming)

[SV88] G. Shackelford and D. Volper. Learning k-DNF with noise in the attributes. In *COLT-88*, 1988.

[Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11), 1984.