Learning Possibilistic Logic Theories from Default Rules

Ondřej Kuželka
Cardiff University, UK
KuzelkaO@cardiff.ac.uk

Jesse Davis
KU Leuven, Belgium
jesse.davis@cs.kuleuven.be

Steven Schockaert
Cardiff University, UK
SchockaertS1@cardiff.ac.uk

Abstract

We introduce a setting for learning possibilistic logic theories from defaults of the form "if alpha then typically beta". We first analyse this problem from the point of view of machine learning theory, determining the VC dimension of possibilistic stratifications as well as the complexity of the associated learning problems, after which we present a heuristic learning algorithm that can easily scale to thousands of defaults. An important property of our approach is that it is inherently able to handle noisy and conflicting sets of defaults. Among others, this allows us to learn possibilistic logic theories from crowdsourced data and to approximate propositional Markov logic networks using heuristic MAP solvers. We present experimental results that demonstrate the effectiveness of this approach.

1 Introduction

Structured information plays an increasingly important role in applications such as information extraction [Dong *et al.*, 2014], question answering [Kalyanpur *et al.*, 2012] and robotics [Beetz *et al.*, 2011]. With the notable exceptions of CYC and WordNet, most of the knowledge bases that are used in such applications have at least partially been obtained using some form of crowdsourcing (e.g. Freebase, Wikidata, ConceptNet). To date, such knowledge bases are mostly limited to facts (e.g. Obama is the current president of the US) and simple taxonomic relationships (e.g. every president is a human). One of the main barriers to crowdsourcing more complex domain theories is that most users are not trained in logic. This is exacerbated by the fact that often (commonsense) domain knowledge is easiest to formalize as defaults (e.g. birds typically fly), and, even for non-monotonic reasoning (NMR) experts, it can be challenging to formulate sets of default rules without introducing inconsistencies (w.r.t. a given NMR semantics) or unintended consequences.

In this paper, we propose a method for learning consistent domain theories from crowdsourced examples of defaults and non-defaults. Since these examples are provided by different users, who may only have an intuitive understanding of the semantics of defaults, together they will typically be inconsistent. The problem we consider is to construct a set of defaults which is consistent w.r.t. the System P semantics [Kraus *et al.*, 1990], and which entails as many of the given defaults and as few of the non-defaults as possible. Taking advantage of the relation between System P and possibilistic logic [Benferhat *et al.*, 1997], we treat this as a learning problem, in which we need to select and stratify a set of propositional formulas.

The contributions of this paper are as follows. First, we show that the problem of deciding whether a possibilistic logic theory exists that perfectly covers all positive and negative examples is Σ_2^P-complete. Second, we formally study the problem of learning from defaults in a standard learning theory setting and we determine the corresponding VC-dimension, which allows us to derive theoretical bounds on how much training data we need, on average, to obtain a system that can classify defaults as being valid or invalid with a given accuracy level. Third, we introduce a heuristic algorithm for learning possibilistic logic theories from defaults and non-defaults. To the best of our knowledge, our method is the first that can learn a consistent logical theory from a set of noisy defaults. We evaluate the performance of this algorithm in two crowdsourcing experiments. In addition, we show how it can be used for approximating maximum a posteriori (MAP) inference in propositional Markov logic networks [Richardson and Domingos, 2006; Dupin de Saint-Cyr *et al.*, 1994]. An online appendix to this paper with additional details is available.[1]

2 Related work

Reasoning with defaults of the form "if α then typically β", denoted as $\alpha \mathrel{|\!\sim} \beta$, has been widely studied [Kraus *et al.*, 1990; Pearl, 1990; Lehmann and Magidor, 1992; Geffner and Pearl, 1992; Goldszmidt *et al.*, 1993; Benferhat *et al.*, 1997]. A central problem in this context is to determine what other defaults can be derived from a given input set. Note, however, that the existing approaches for reasoning about default rules all require some form of consistency (e.g. the input set cannot contain both $a \mathrel{|\!\sim} b$ and $a \mathrel{|\!\sim} \neg b$). As a result, these approaches cannot directly be used for reasoning about noisy crowdsourced defaults.

To the best of our knowledge, this is the first paper that considers a machine learning setting where the input consists of default rules. Several authors have proposed ap-

[1] http://arxiv.org/abs/1604.05273

proaches for constructing possibility distributions from data; see [Dubois and Prade, 2015] for a recent survey. However, such methods are generally not practical for constructing possibilistic logic theories. The possibilistic counterpart of the Z-ranking constructs a possibilistic logic theory from a set of defaults, but it requires that these defaults are consistent and cannot handle non-defaults [Benferhat *et al.*, 1997], although an extension of the Z-ranking that can cope with non-defaults was proposed in [Booth and Paris, 1998]. Some authors have also looked at the problem of learning sets of defaults from data [Benferhat *et al.*, 2003; Kern-Isberner *et al.*, 2008], but the performance of these methods has not been experimentally tested. In [Serrurier and Prade, 2007], a possibilistic inductive logic programming (ILP) system is proposed, which uses a variant of possibilistic logic for learning rules with exceptions. However, as is common for ILP systems, this method only considers classification problems, and cannot readily be applied to learn general possibilistic logic theories. Our work can also be seen as conceptually related to belief merging, although we are not aware of any existing methods for merging inconsistent sets of default rules. Finally note that the setting of learning from default rules as introduced in this paper can be seen as a non-monotonic counterpart of an ILP setting called *learning from entailment* [De Raedt, 1997].

3 Background

3.1 Possibilistic logic

A stratification of a propositional theory $\mathcal{T}$ is an ordered partition of the set of formulas in $\mathcal{T}$. A theory in possibilistic logic [Dubois *et al.*, 1994] is a set of formulas of the form (α, λ), with α a propositional formula and $\lambda \in]0, 1]$ a certainty weight. These certainty weights are interpreted in a purely ordinal fashion, hence a possibilistic logic theory is essentially a stratification of a propositional theory. The strict λ-cut $\Theta_{\overline{\lambda}}$ of a possibilistic logic theory Θ is defined as $\Theta_{\overline{\lambda}} = \{\alpha \,|\, (\alpha, \mu) \in \Theta, \mu > \lambda\}$. The inconsistency level *inc*(Θ) of Θ is the lowest certainty level λ in $[0, 1]$ for which the classical theory $\Theta_{\overline{\lambda}}$ is consistent. An inconsistency-tolerant inference relation $\vdash_{poss}$ for possibilistic logic can then be defined as follows:

$$\Theta \vdash_{poss} \alpha \quad \text{iff} \quad \Theta_{\overline{inc(\Theta)}} \models \alpha$$

We will write $(\Theta, \alpha) \vdash_{poss} \beta$ as an abbreviation for $\Theta \cup \{(\alpha, 1)\} \vdash_{poss} \beta$. It can be shown that $\Theta \vdash_{poss} (\alpha, \lambda)$ can be decided by making $O(\log_2 k)$ calls to a SAT solver, with k the number of certainly levels in Θ [Lang, 2001].

There is a close relationship between possibilistic logic and the rational closure of a set of defaults. Recall that $\alpha \,|\!\sim \beta$ is tolerated by a set of defaults $\{\alpha_1 \,|\!\sim \beta_1, ..., \alpha_n \,|\!\sim \beta_n\}$ if the classical formula $\alpha \wedge \beta \wedge \bigwedge_i (\neg\alpha_i \vee \beta_i)$ is consistent [Pearl, 1990]. Let Δ be a set of defaults. The rational closure of Δ is based on a stratification $\Delta_1, ..., \Delta_k$, known as the Z-ordering, where each Δ_j contains all defaults from $\Delta \setminus (\Delta_1 \cup ... \Delta_{j-1})$ which are tolerated by $\Delta \setminus (\Delta_1 \cup ... \cup \Delta_{j-1})$. Intuitively, Δ_1 contains the most general default rules, Δ_2 contains exceptions to these rules, Δ_3 contains exceptions to these exceptions, etc. Given the stratification $\Delta_1, ..., \Delta_k$ we define the possibilistic logic theory $\Theta = \{(\neg\alpha \vee \beta, \lambda_i) \,|\, (\alpha \,|\!\sim \beta) \in \Delta_i\}$, where we assume $0 < \lambda_1 < ... < \lambda_k \leq 1$. It then holds that $\alpha \,|\!\sim \beta$ is in the rational closure of Δ iff $(\Theta, \alpha) \vdash_{poss} \beta$ [Benferhat *et al.*, 1998].

3.2 Learning Theory

We now cover some basic notions from statistical learning theory [Vapnik, 1995]. We restrict ourselves to binary classification problems, where the two labels are 1 and -1. Let $\mathcal{X}$ be a set of *examples*. A *hypothesis* is a function $h : \mathcal{X} \to \{-1, 1\}$. A hypothesis h is said to cover an example $e \in \mathcal{X}$ if $h(e) = 1$. Consider a set $\mathcal{S} \subseteq \mathcal{X} \times \{-1, 1\}$ of n labeled examples that have been iid sampled from a distribution p. A hypothesis h's sample error rate is $err_{\mathcal{S}}(h, \mathcal{S}) = \frac{1}{n} \sum_{(x,c)\in\mathcal{S}} \mathbf{1}(h(x) \neq c)$ where $\mathbf{1}(\alpha) = 1$ if $\alpha \equiv$ *true* and $\mathbf{1}(\alpha) = 0$ otherwise. A hypothesis h's expected error w.r.t. the probability distribution p is given by $err_p(h) = \mathbf{E}_{(X,C)\sim p}[\mathbf{1}(h(X) \neq C)]$. Statistical learning theory provides tools for bounding the probability $P(\sup_{h\in\mathcal{H}} |err_p(h) - err_{\mathcal{S}}(h, \mathcal{S})| \geq \epsilon)$, where $\mathcal{S}$ is known to be sampled iid from p but p itself is unknown. These bounds link h's training set error to its (probable) performance on other examples drawn from the same distribution, and therefore permits theoretically controlling overfitting. The most important bounds of this type depend on the Vapnik-Chervonenkis (VC) dimension [Vapnik, 1995].

Definition 1 (Vapnik-Chervonenkis (VC) dimension)**.** *A hypothesis set $\mathcal{H}$ is said to shatter a set of examples $\mathcal{Y}$ if for every subset $\mathcal{Z} \subseteq \mathcal{Y}$ there is a hypothesis $h \in \mathcal{H}$ such that $h(e) = 1$ for every $e \in \mathcal{Z}$ and $h(e) = -1$ for every $e \in \mathcal{Y} \setminus \mathcal{Z}$. The VC dimension of $\mathcal{H}$ is the cardinality of the largest set that is shattered by $\mathcal{H}$.*

Upper bounds based on the VC dimension are increasing functions of the VC dimension and decreasing functions of the number of examples in the training sample $\mathcal{S}$. Ideally, the goal is to minimize expected error, but this cannot be evaluated since p is unknown. *Structural risk minimization* [Vapnik, 1995] helps with this if the hypothesis set can be organized into a hierarchy of nested hypothesis classes of increasing VC dimension. It suggests selecting hypotheses that minimize a risk composed of the training set error and a complexity term, e.g. if two hypotheses have the same training set error, the one originating from the class with lower VC dimension should be preferred.

4 Learning from Default Rules

In this section, we formally describe a new learning setting for possibilistic logic called *learning from default rules*. We assume a finite alphabet Σ is given. An example is a default rule over Σ and a hypothesis is a possibilistic logic theory over Σ. A hypothesis h predicts the class of an example $e = \alpha \,|\!\sim \beta$ by checking if h covers e, in the following sense.

Definition 2 (Covering)**.** *A hypothesis $h \in \mathcal{H}$* covers *an example $e = \alpha \,|\!\sim \beta$ if $(h, \alpha) \vdash_{poss} \beta$.*

The hypothesis h predicts positive, i.e. $h(\alpha \,|\!\sim \beta) = 1$, iff h covers e, and else predicts negative, i.e. $h(\alpha \,|\!\sim \beta) = -1$.

Example 1. *Let us consider the following set of examples*

$$\mathcal{S} = \{(bird \wedge antarctic \mid\sim \neg flies, 1), (bird \mid\sim \neg flies, -1)\}$$

The following hypotheses over the alphabet $\{bird, flies, antarctic\}$ *cover all positive and no negative examples:*

$$h_1 = \{(bird, 1), (antarctic \rightarrow \neg flies, 1)\}$$
$$h_2 = \{(flies, 0.5), (antarctic \rightarrow \neg flies, 1)\}$$
$$h_3 = \{(antarctic \rightarrow \neg flies, 1)\}$$

The learning task can be formally described as follows.

Given: A multi-set $\mathcal{S}$ which is an i.i.d. sample from a set of default rules over a given finite alphabet Σ.

Do: Learn a possiblistic logic theory that covers all positive examples and none of the negative examples in $\mathcal{S}$.

The above definition assumes that $\mathcal{S}$ is perfectly separable, i.e. it is possible to perfectly distinguish positive examples from negative examples. In practice, we often relax this requirement, and instead aim to find a theory that minimizes the training set error. Similar to learning in graphical models, this learning task can be decomposed into *parameter learning* and *structure learning*. In our context, the goal of parameter learning is to convert a set of propositional formulas into a possibilistic logic theory, while the goal of structure learning is to decide what that set of propositional formulas should be.

4.1 Parameter Learning

Parameter learning assumes that the formulas of the possibilistic logic theory are fixed, and only the certainty weights need to be assigned. As the exact numerical values of the certainty weights are irrelevant, we will treat parameter learning as the process of finding the most suitable stratification of a given set of formulas, e.g. the one which minimizes training error or structural risk (cf. Section 4.2).

Example 2. *Let* $\mathcal{S} = \{(penguin \mid\sim bird, 1), (bird \mid\sim flies, 1), (penguin \mid\sim \neg flies, 1), (\mid\sim bird, -1), (bird \mid\sim penguin, -1)\}$ *and* $\mathcal{T} = \{bird, flies, penguin, \neg penguin \vee \neg flies\}$. *A stratification of* $\mathcal{T}$ *which minimizes the training error on the examples from* $\mathcal{S}$ *is* $\mathcal{T}^* = \{(bird, 0.25), (penguin, 0.25), (flies, 0.5), (\neg penguin \vee \neg flies, 1)\}$ *which is equivalent to* $\mathcal{T}^{**} = \{(flies, 0.5), (\neg penguin \vee \neg flies, 1)\}$ *because* $inc(\mathcal{T}^*) = 0.25$. *Note that* $\mathcal{T}^{**}$ *correctly classifies all examples except* $(penguin \mid\sim bird, 1)$.

Given a set of examples $\mathcal{S}$, we write $\mathcal{S}^+ = \{\alpha | (\alpha, 1) \in \mathcal{S}\}$ and $\mathcal{S}^- = \{\alpha | (\alpha, -1) \in \mathcal{S}\}$). A stratification $\mathcal{T}^*$ of a theory $\mathcal{T}$ is a *separating stratification* of $\mathcal{S}^+$ and $\mathcal{S}^-$ if it covers all examples from $\mathcal{S}^+$ and no examples from $\mathcal{S}^-$.

Example 3. *Let us consider the following set of examples* $\mathcal{S} = \{(\mid\sim \neg x, 1), (\mid\sim \neg y, 1), (x \mid\sim a, 1), (y \mid\sim b, 1), (x \wedge y \mid\sim a, -1)\}$. *Let* $\mathcal{T} = \{\neg x, \neg y, \neg x \vee a, \neg y \vee b\}$. *The following stratification is a separating stratification of* $\mathcal{S}^+$ *and* $\mathcal{S}^-$: $h = \{(\neg x, 0.25), (\neg x \vee a, 0.5), (\neg y, 0.75), (\neg y \vee b, 1)\}$. *Note that the Z-ranking of* $\mathcal{S}^+$ *also corresponds to a stratification of* $\mathcal{T}$, *as* $\mathcal{T}$ *contains exactly the clause representations of the positive examples. However using the Z-ranking leads to a different stratification, which is:* $h_z = \{(\neg x, 0.5), (\neg y, 0.5), (\neg x \vee a, 1), (\neg y \vee b, 1)\}$. *Note that* $h_z(x \wedge y \mid\sim a) = 1$ *whereas* $h(x \wedge y \mid\sim a) = -1$.

Because arbitrary stratifications can be chosen, there is substantial freedom to ensure that negative examples are not covered. This is true even when the set of considered formulas is restricted to the clause representations of the positive examples, as seen in Example 3. Unfortunately, the problem of finding an optimal stratification is computationally hard.

Theorem 1. *Deciding whether a separating stratification exists for given* $\mathcal{T}$, $\mathcal{S}^+$ *and* $\mathcal{S}^-$ *is a* Σ_2^P*-complete problem.*

Proof. The proof of the membership result is trivial. We show the hardness result by reduction from the Σ_2^P-complete problem of deciding the satisfiability of quantified Boolean formulas of the form $\exists X \forall Y : \Phi(X, Y)$ where X and Y are vectors of propositional variables and $\Phi(X, Y)$ is a propositional formula. Let $\mathcal{T} = X \cup \{\neg x : x \in X\} \cup \{\Phi(X, Y) \rightarrow aux\}$ be a propositional theory, let $\mathcal{S}^+ = \{\mid\sim aux\}$ and $\mathcal{S}^- = \emptyset$. We need to show that $\exists X \forall Y : \Phi(X, Y)$ is satisfiable if and only if there exists a separating stratification for $\mathcal{T}$, $\mathcal{S}^+$ and $\mathcal{S}^-$. ($\Rightarrow$) Let θ be an assignment of variables in X such that $\forall Y : \Phi(X\theta, Y)$ is true. Then we can construct the separating stratification as

$$(\{\Phi(X, Y) \rightarrow aux\} \cup \{x \in X : x\theta = 1\} \cup \{\neg x : x \in X \text{ and } x\theta = 0\},$$
$$\{\neg x : x \in X \text{ and } x\theta = 1\} \cup \{x \in X : x\theta = 0\}).$$

Since $\Phi(X, Y)$ will always be true in any model consistent with the highest level of the stratification, because of the way we chose x and $\neg x$ for this level, so will *aux*. ($\Leftarrow$) Let $\mathcal{T}^*$ be a stratification of $\mathcal{T}$ which entails the default rule $\mid\sim aux$. We can assume w.l.o.g. that $\mathcal{T}^*$ has only two levels. Since $\mathcal{T}^*$ is a separating stratification, we must have $\mathcal{T}^* \vdash_{poss} aux$. Therefore the highest level L^* of $\mathcal{T}^*$ must be a consistent theory and $\Phi(X, Y)$ must be true in all of its models. Let $X' = \{x \in X : x \in L^* \text{ or } \neg x \in L^*\}$ and $X'' = X \setminus X'$. We can construct an assignment θ to variables in X' by setting $x\theta = 1$ for $x \in L^*$ and $x\theta = 0$ for $\neg x \in L^*$. It follows from the construction that $\forall X'' \forall Y : \Phi(X\theta, Y)$ must be true. □

As this result reveals, in practice we will need to rely on heuristic methods for parameter learning. In Section 4.3 we will propose such a heuristic method, which will moreover also include structure learning.

4.2 VC Dimension of Possibilistic Logic Theories

We explore the VC dimension of the set of possible stratifications of a propositional theory, as this will allow us to provide probabilistic bounds on the generalization ability of a learned possibilistic logic theory. Let us write $Strat(\mathcal{T})$ for the set of all stratifications of a propositional theory $\mathcal{T}$, and let $Strat^{(k)}(\mathcal{T})$ be the set of all stratifications with at most k levels. The following proposition provides an upper bound for the VC dimension and can be proved by bounding the cardinality of $Strat^{(k)}(\mathcal{T})$.

Proposition 1. *Let* $\mathcal{T}$ *be a set of* n *propositional formulas. Then* $VC(Strat^{(k)}(\mathcal{T})) \leq n \log_2 k$.

In the next theorem, we establish a lower bound on the VC dimension of stratifications with at most k levels which shows that the above upper bound is asymptotically tight.

Theorem 2. *For every k, n, $k \leq n$, there is a propositional theory $\mathcal{T}$ consisting of n formulas such that*

$$VC(\textit{Strat}^{(k)}(\mathcal{T})) \geq \frac{1}{4} n(\log_2 k - 1).$$

To prove Theorem 2, we need the following lemmas; some straightforward proofs are omitted due to space constraints.

Lemma 1. *If $\mathcal{S}$ is a totally ordered set, let $\textit{kth}(\mathcal{S}, i)$ denote the i-th highest element of $\mathcal{S}$. Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be a set of cardinality $n = 2^m$ where $m \in \mathbb{N} \setminus \{0\}$. Let*

$$\begin{aligned}
\mathcal{C} = \{&x_1 < x_2, x_3 < x_4, \ldots, x_{n-1} < x_n,\\
&\textit{kth}(\{x_1, x_2\}, 1) < \textit{kth}(\{x_3, x_4\}, 1),\\
&\textit{kth}(\{x_1, x_2\}, 2) < \textit{kth}(\{x_3, x_4\}, 2),\\
&\textit{kth}(\{x_5, x_6\}, 1) < \textit{kth}(\{x_7, x_8\}, 1),\\
&\textit{kth}(\{x_5, x_6\}, 2) < \textit{kth}(\{x_7, x_8\}, 2),\\
&\ldots\\
&\textit{kth}(\{x_1, x_2, x_3, x_4\}, 1) < \textit{kth}(\{x_5, x_6, x_7, x_8\}, 1),\\
&\textit{kth}(\{x_1, x_2, x_3, x_4\}, 2) < \textit{kth}(\{x_5, x_6, x_7, x_8\}, 2),\\
&\textit{kth}(\{x_1, x_2, x_3, x_4\}, 3) < \textit{kth}(\{x_5, x_6, x_7, x_8\}, 3),\\
&\textit{kth}(\{x_1, x_2, x_3, x_4\}, 4) < \textit{kth}(\{x_5, x_6, x_7, x_8\}, 4),\\
&\ldots\\
&\textit{kth}(\{x_1, \ldots, x_{n/2}\}, n/2) < \textit{kth}(\{x_{n/2+1}, \ldots, x_n\}, n/2)\}
\end{aligned}$$

be a set of $\frac{1}{2} n \log_2 n$ inequalities. Then for any $\mathcal{C}' \subseteq \mathcal{C}$ there is a permutation of $\mathcal{X}$ satisfying all constraints from $\mathcal{C}'$ and no constraints from $\mathcal{C} \setminus \mathcal{C}'$.

Lemma 2. *Let $\textit{at-least}_k(x_1, x_2, \ldots, x_n)$ denote a Boolean formula which is true if and only if at least k of the arguments are true. Let $\mathcal{X} = \{x_1, \ldots, x_m\}$ be a set of propositional logic variables and $\pi = (x_{i_1}, x_{i_2}, \ldots, x_{i_m})$ be a permutation of elements from $\mathcal{X}$. Let $0 \leq k \leq \min\{m_y, m_z\}$. Let*

$$\mathcal{T}^* = \{(x_{i_1}, 1/m), (x_{i_2}, 1/(m-1)), \ldots, (x_{i_m}, 1)\}$$

be a possibilistic logic theory. Let $\mathcal{Y} = \{y_1, \ldots, y_{m_y}\}$ and $\mathcal{Z} = \{z_1, \ldots, z_{m_z}\}$ be disjoint subsets of $\mathcal{X}$. Then

$$\begin{aligned}
(\mathcal{T}^*, &\textit{at-least}_{m_y - k + 1}(\neg y_1, \neg y_2, \ldots, \neg y_{m_y})) \vdash_{poss}\\
&\textit{at-least}_k(z_1, z_2, \ldots, z_{m_z})
\end{aligned}$$

iff $\textit{kth}(\{y_1, \ldots, y_{m_y}\}, k) < \textit{kth}(\{z_1, \ldots, z_{m_z}\}, k)$ w.r.t. the ordering given by the permutation π.

Lemma 3. *For every $n = 2^m$ there is a propositional theory $\mathcal{T}$ consisting of n formulas such that*

$$VC(\textit{Strat}(\mathcal{T})) \geq \frac{1}{2} n \log_2 n.$$

Proof. Let $\mathcal{T} = \{x_1, \ldots, x_n\}$ be a set of propositional variables where $n = 2^m$, $m \in N \setminus \{0\}$ and let $\mathcal{C}$ be defined as in Lemma 1. Let $\mathcal{D} =$

$$\begin{aligned}
\{&\textit{at-least}_{l-k+1}(\neg x_{i_1}, \ldots, \neg x_{i_l}) \mid\sim \textit{at-least}_k(x_{j_1}, \ldots, x_{j_l}) |\\
&(\textit{kth}(\{x_{i_1}, \ldots, x_{i_l}\}, k) < \textit{kth}(\{x_{j_1}, \ldots, x_{j_l}\}, k)) \in \mathcal{C}\},
\end{aligned}$$

i.e. $\mathcal{D}$ contains one default rule for every inequality from $\mathcal{C}$. It follows from Lemma 1 and Lemma 2 that the set $\mathcal{D}$ can be shattered by stratifications of the propositional theory $\mathcal{T}$. The cardinality of $\mathcal{D}$ is $\frac{1}{2} n \log_2 n$. Therefore the VC dimension of stratifications of $\mathcal{T}$ is at least $\frac{1}{2} n \log_2 n$. □

Proof of Theorem 2. We show that if k and n are powers of two then $\frac{1}{2} n \log_2 k$ is a lower bound of the VC dimension. The general case of the theorem then follows straightforwardly. Let $\mathcal{T} = \bigcup_{i=1}^{\frac{n}{k}} \{x_{(i-1)\cdot k+1}, \ldots, x_{i \cdot k}\}$ and let $\mathcal{D}_i$ be a set of default rules of cardinality $\frac{1}{2} k \log_2 k$ shattered by $\textit{Strat}(\{x_{(i-1)\cdot k+1}, \ldots, x_{i \cdot k}\})$. It follows from Lemma 3 that such a set $\mathcal{D}_i$ always exists. Let $\mathcal{D} = \bigcup_{i=1}^{\frac{n}{k}} \mathcal{D}_i$. Then $\mathcal{D}$ has cardinality $\frac{1}{2} n \log_2 k$ and is shattered by $\textit{Strat}^{(k)}(\mathcal{T})$. To see that the latter holds, note that the sets of formulas $\textit{Strat}(\{x_{(i-1)\cdot k+1}, \ldots, x_{i \cdot k}\})$ are disjoint. Therefore, if we want to find a stratification from $\textit{Strat}^{(k)}(\mathcal{T})$ which covers only examples from an arbitrary set $\mathcal{D}' \subseteq \mathcal{D}$ and no other examples from $\mathcal{D}$ then we can merge stratifications of $\{x_{(i-1)\cdot k+1}, \ldots, x_{i \cdot k}\}$ which cover exactly the examples from $\mathcal{D}_i \cap \mathcal{D}'$, where merging stratifications is done by levelwise unions. □

Combining the derived lower bounds and upper bounds on the VC dimension together with the structural risk minimization principle, we find that given two stratifications with the same training set error rate, we should prefer the one with the fewest levels. Furthermore, when structure learning is used, it is desirable for learned theories to be compact. A natural learning problem then consists of selecting a small subset of $\mathcal{T}$, where $\mathcal{T}$ corresponds to the set of formulas considered by the structure learner, and identifying a stratification only for that subset. The results in this section can readily be extended to provide bounds on the VC dimension of this problem. Let $\mathcal{T}$ be a propositional theory of cardinality n and let $m < n$ be a positive integer. The VC dimension of the set of hypotheses involving at most m formulas from $\mathcal{T}$ and having at most k levels is bounded by $m(\log_2 n + \log_2 k)$. This can simply be obtained by upper-bounding the number of the different stratifications with at most k levels and m formulas selected from a set of cardinality n, by $n^m \cdot k^m$.

4.3 Heuristic Learning Algorithm

In this section, we propose a practical heuristic algorithm for learning a possibilistic logic theory from a set $\mathcal{S}$ of positive and negative examples of default rules. Our method combines greedy structure learning with greedy weight learning. We assume that every default or non-default $\alpha \mid\sim \beta$ in $\mathcal{S}$ is such that $\neg\alpha$ and β correspond to clauses.

The algorithm starts by initializing the "working" stratification $\mathcal{T}^*$ to be an empty list. Then it repeats the following revision procedure for a user-defined number of iterations n, or until a timeout is reached. First, it generates a set of candidate propositional clauses C as follows:

- It samples a set of defaults $\alpha \mid\sim \beta$ from the examples that are misclassified by $\mathcal{T}^*$.
- For each default $\alpha \mid\sim \beta$ which has been sampled, it samples a subclause $\neg\alpha'$ of $\neg\alpha$ and a subclause β' of β. If $\alpha \mid\sim \beta$ is a positive example then $\neg\alpha' \vee \beta'$ is added to C; if it is a negative example, then $\neg\alpha' \vee \beta''$ is added instead, where β'' is obtained from β' by negating each of the literals.

The algorithm then tries to add each formula in C to an existing level of $\mathcal{T}^*$ or to a newly inserted level. It picks the clause c whose addition leads to the highest accuracy and adds it to $\mathcal{T}^*$. The other clauses from C are discarded. In case of ties, the clause which leads to the stratification with the fewest levels is selected, in accordance with the structural risk minimization principle and our derived VC dimension. If there are multiple such clauses, then it selects the shortest among them. Subsequently, the algorithm tries to greedily minimize the newly added clause c, by repeatedly removing literals as long as this does not lead to an increase in the training set error. Next, the algorithm tries to revise $\mathcal{T}^*$ by greedily removing clauses whose deletion does not increase the training set error. Finally, as the last step of each iteration, the weights of all clauses are optimized by greedily reinserting each clause in the theory.

5 Experiments

We evaluate our heuristic learning algorithm[2] in two different applications: learning domain theories from crowdsourced default rules and approximating MAP inference in propositional Markov logic networks. As we are not aware of any existing methods that can learn a consistent logical theory from a set of noisy defaults, there are no baseline methods to which our method can directly be compared. However, if we fix a target literal l, we can train standard classifiers to predict for each propositional context α whether the default $\alpha \mid\!\sim l$ holds. This can only be done consistently with "parallel" rules, where the literals in the consequent do not appear in antecedents. We will thus compare our method to three traditional classifiers on two crowdsourced datasets of parallel rules: random forests [Breiman, 2001], C4.5 decision trees [Quinlan, 1993], and the rule learner RIPPER [Cohen, 1995]. Random forests achieve state-of-the-art accuracy[3] but its models are difficult to interpret. Decision trees are often less accurate but more interpretable than random forests. Finally, rule learners have the most interpretable models, but often at the expense of lower accuracy. In the second experiment, approximating MAP inference, we do not restrict ourselves to parallel rules. In this case, only our method can guarantee that the predicted defaults will be consistent.

5.1 Methodology

Our learning algorithm is implemented in Java and uses the SAT4j library [Berre and Parrain, 2010]. The implementation contains a number of optimizations which make it possible to handle datasets of thousands of default rules, including caching, parallelization, detection of relevant possibilistic subtheories for deciding entailment queries and unit propagation in the possibilistic logic theories.

We use the Weka [Hall *et al.*, 2009] implementations for the three baselines. When using our heuristic learning algorithm, we run it for a maximum time of 10 hours for the crowdsourcing experiments reported in Section 5.2 and for one hour for the experiments reported in Section 5.3. For C4.5 and RIPPER, we use the default settings. For random forests, we used the default settings and set the number of trees to 100.

5.2 Learning from Crowdsourced Examples

We used CrowdFlower, an online crowdsourcing platform, to collect expert rules about two domains. In the first experiment, we created 3706 scenarios for a team on offense in American football by varying the field position, down and distance, time left, and score difference. Then we presented six choices for a play call (punt, field goal, run, pass, kneel down, do not know/it depends) and asked the user to select the most appropriate one. All scenarios were presented to 5 annotators. A manual inspection of a subset of the rules revealed that they are of reasonably high quality. In a second experiment, users were presented with 2388 scenarios based on Texas hold'em poker situations, where users were asked whether in a given situation they would typically fold, call or raise, with a fourth option again being "do not know/it depends". Each scenario was again presented to 5 annotators. Given the highly subjective nature of poker strategy, it was not possible to enforce the usual quality control mechanism on CrowdFlower in this case, and the quality of the collected rules was accordingly found to be more variable.

In both cases, the positive examples are the rules obtained via crowdsourcing, while negative examples are created by taking positive examples and randomly selecting a different consequent. To create training and testing sets, we divided the data based on annotator ID so that all rules labeled by a given annotated appear only in the training set or only in the testing set, to prevent leakage of information. We added a set of hard rules to the possibilistic logic theories to enforce that only one choice should be selected for a game situation. The baseline methods were presented with the same information, in the sense that the problem was presented as a multi-class classification problem, i.e. given a game situation, the different algorithms were used to predict the most typical action (with one additional option being that none of the actions is typical). The results are summarized in Table 1.

In the poker experiment, our approach obtained slightly higher accuracy than random forest and RIPPER but performed slightly worse than C4.5. However, a manual inspection showed that a meaningful theory about poker strategy was learned. For example, at the lowest level, the possibilistic logic theory contains the rule "call", which makes sense given the nature of the presented scenarios. At a higher level, it contains more specific rules such as "if you have three of a kind then raise". At the level above, it contains exceptions to these more specific rules such as "If you have three of a kind, there are three hearts on the board and your opponent raised on the river then call".

In the American football experiment, our approach obtained lower accuracy than the competing algorithms. The best accuracy was achieved by C4.5. Again, we also manually inspected the learned possibilistic logic theory and found that it captures some general intuitions and known strategy about the game. For example, the most general rule is "pass"

[2]The data, code, and learned models are available from https://github.com/supertweety/.

[3]A recent large-scale empirical evaluation has shown that variants of the random forest algorithm tend to perform best on real-life datasets [Fernández-Delgado *et al.*, 2014].

	Poss.	Rand. F.	C4.5	RIPPER
Poker	40.5	38.6	**41.1**	39.9
Football	68.3	72.4	**74.6**	73.1
NLTCS	**78.1**	69.6	70.2	67.7
MSNBC	**62.0**	61.9	**62.0**	48.8
Plants	73.1	**77.8**	71.4	53.8
DNA	52.8	**56.6**	54.9	51.1

Table 1: Test set accuracies.

which is the most common play type. Another example is that second most general level has several rules that say on fourth down and long you should punt. More specific levels allow for cases when you should not punt on fourth down, such as when you are in field goal range.

Despite not achieving the same accuracy as C4.5 in this experiment, it nonetheless seems that our method is useful for building up domain theories by crowdsourcing opinions. The learned domain theories are easy to interpret (e.g., the size of the poker theory, as a sum of rule lengths, is more than 10 times smaller than the number of nodes in the learned tree) and capture relevant strategies for both games. The models obtained by classifiers such as C4.5, on the other hand, are often difficult to interpret. Moreover, traditional classifiers such as C4.5 can only be applied to parallel rules, and will typically lead to inconsistent logical theories in more complex domains. In contrast, our method can cope with arbitrary default rules as input, making it much more broadly applicable for learning domain theories.

5.3 Approximating MAP Inference

Markov logic networks can be seen as weighted logical theories. The weights assigned to formulas are intuitively seen as penalties; they are used to induce a probability distribution over the set of possible world. Here we are interested in maximum a posteriori (MAP) inference. Specifically, we consider the following entailment relation from [Dupin de Saint-Cyr *et al.*, 1994]: $(\mathcal{M},\alpha) \vdash_{MAP} \beta$ iff $\forall \omega \in \max(\mathcal{M},\alpha) : \omega \models \beta$ where $\mathcal{M}$ is an MLN, α and β are propositional formulas and $\max(\mathcal{M},\alpha)$ is the set of most probable models of α, w.r.t. the probability distribution induced by $\mathcal{M}$. Note that MAP inference only depends on an ordering of possible worlds. It was shown in [Kuželka *et al.*, 2015] that for every propositional MLN $\mathcal{M}$ there exists a possibilistic logic theory Θ such that $(\mathcal{M},\alpha) \vdash_{MAP} \beta$ iff $(\Theta,\alpha) \vdash_{poss} \beta$. Such a translation can be useful in practice, as possibilistic logic theories tend to be much easier to interpret, given that the weights associated with different formulas in an MLN can interact in non-trivial ways. Unfortunately, in general Θ is exponentially larger than $\mathcal{M}$. Moreover, the translation from [Kuželka *et al.*, 2015] requires an exact MAP solver, whereas most such solvers are approximate.

Therefore, rather than trying to capture MAP inference exactly, here we propose to learn a possibilistic logic theory from a set of examples of valid and invalid MAP entailments $(\mathcal{M},\alpha) \vdash_{MAP} \beta$. Since our learning algorithm can handle non-separable data, we can use approximate MAP solvers for generating these examples, which leads to further gains in scalability. As is common, the evidence α consists of conjunctions of up to k literals, and the conclusion β consists of an individual literal. To create examples, we randomly generate a large number of evidence formulas α, each time considering a large number of possible βs. If β is MAP-entailed by α, we add $\alpha \mid\sim \beta$ to the set of positive examples; otherwise we add it to the set of negative examples. Notice that defaults in these experiments are not restricted to be just "parallel" rules.

We considered propositional MLNs learned from NLTCS, MSNBC, Plants and DNA data using the method from [Lowd and Davis, 2014]. These are standard datasets, and have 16, 17, 69 and 180 Boolean random variables, respectively. We used the existing train/tune/test division of the data. For each dataset, we generated 1000 training examples and 1000 testing examples of default rules as described above, and considered evidence formulas α of up 5 literals. We learn the possibilistic logic theory on the training examples, and report results on the held-out testing examples. To use the classical learners, we represent the antecedent using two Boolean attributes for each variable in the domain: the first indicates the variable's positive presence in the antecedent while the second indicates its negative presence. We represent the consequent in the same way. The label of an example is positive if it appears in the default theory and negative otherwise. While this allows us to predict whether a default $a \mid\sim b$ should be true, the set of defaults predicted by the classical methods will in general not be consistent.

The last four rows of Table 1 show the test set accuracy for each approach on each domain. Overall, the learned possibilistic logic theories have similar performance to the decision tree and random forest models, and outperform RIPPER. This is quite remarkable, as the possibilistic logic theories are much more interpretable (containing approximately 50% fewer literals than the decision trees), which usually means that we have to accept a lower accuracy. Moreover, while the other methods can also be used for predicting MAP entailment, only our method results in a consistent logical theory, which could e.g. easily be combined with expert knowledge.

6 Conclusions

The aim of this paper was to study the problem of reasoning with default rules from a machine learning perspective. We have formally introduced the problem of learning from defaults and have analyzed its theoretical properties. Among others, we have shown that the complexity of the main decision problem is Σ_2^P complete, and we have established asymptotically tight bounds on the VC-dimension. At the practical level, we have proposed practical heuristic learning algorithm, which can scale to datasets with thousands of rules. We have presented experimental results that show the application potential of the proposed learning algorithm, considering two different application settings: learning domain theories by crowdsourcing expert opinions and approximating propositional MLNs. We believe that the methods proposed in this paper will open the door to a wider range of applications of default reasoning, where we see defaults as a convenient interface between experts and learned domain models.

Acknowledgments

This work has been supported by a grant from the Leverhulme Trust (RPG-2014-164). Jesse Davis is partially supported by the KU Leuven Research Fund (C22/15/015), and FWO-Vlaanderen (G.0356.12, SBO-150033).

References

[Beetz *et al.*, 2011] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mosenlechner, D. Pangercic, T. Ruhr, and M. Tenorth. Robotic roommates making pancakes. In *Proceedings of the 11th IEEE-RAS International Conference on Humanoid Robots*, pages 529–536, 2011.

[Benferhat *et al.*, 1997] S. Benferhat, D. Dubois, and H. Prade. Nonmonotonic reasoning, conditional objects and possibility theory. *Artificial Intelligence*, 92(1-2):259–276, 1997.

[Benferhat *et al.*, 1998] S. Benferhat, D. Dubois, and H. Prade. Practical handling of exception-tainted rules and independence information in possibilistic logic. *Applied intelligence*, 9(2):101–127, 1998.

[Benferhat *et al.*, 2003] S. Benferhat, D. Dubois, S. Lagrue, and H. Prade. A big-stepped probability approach for discovering default rules. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 11(Supplement-1):1–14, 2003.

[Berre and Parrain, 2010] D. Le Berre and A. Parrain. The SAT4J library, release 2.2. *Journal on Satisfiability, Boolean Modeling and Computation*, 7:50–64, 2010.

[Booth and Paris, 1998] R. Booth and J. B. Paris. A note on the rational closure of knowledge bases with both positive and negative knowledge. *Journal of Logic, Language and Information*, 7:165–190, 1998.

[Breiman, 2001] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[Cohen, 1995] W. W. Cohen. Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123, 1995.

[De Raedt, 1997] L. De Raedt. Logical settings for concept-learning. *Artificial Intelligence*, 95(1):187–201, 1997.

[Dong *et al.*, 2014] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 601–610, 2014.

[Dubois and Prade, 2015] D. Dubois and H. Prade. Practical methods for constructing possibility distributions. *International Journal of Intelligent Systems*, 2015.

[Dubois *et al.*, 1994] D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In D. Nute D. Gabbay, C. Hogger J. Robinson, editor, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 439–513. Oxford University Press, 1994.

[Dupin de Saint-Cyr *et al.*, 1994] F. Dupin de Saint-Cyr, J. Lang, and T. Schiex. Penalty logic and its link with Dempster-Shafer theory. In *Uncertainty in Artificial Intelligence*, pages 204–211, 1994.

[Fernández-Delgado *et al.*, 2014] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we need hundreds of classifiers to solve real world classification problems? *JMLR*, 15:3133–3181, 2014.

[Geffner and Pearl, 1992] H. Geffner and J. Pearl. Conditional entailment: Bridging two approaches to default reasoning. *Artificial Intelligence*, 53(2):209–244, 1992.

[Goldszmidt *et al.*, 1993] Moisés Goldszmidt, Paul Morris, and Judea Pearl. A maximum entropy approach to nonmonotonic reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(3):220–232, 1993.

[Hall *et al.*, 2009] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[Kalyanpur *et al.*, 2012] A. Kalyanpur, B. K. Boguraev, S. Patwardhan, J. W. Murdock, A. Lally, C. Welty, J. M. Prager, B. Coppola, A. Fokoue-Nkoutche, L. Zhang, et al. Structured data and inference in DeepQA. *IBM Journal of Research and Development*, 56(3.4):10–1, 2012.

[Kern-Isberner *et al.*, 2008] G. Kern-Isberner, M. Thimm, and M. Finthammer. Qualitative knowledge discovery. In *Semantics in Data and Knowledge Bases*, pages 77–102. Springer Berlin Heidelberg, 2008.

[Kraus *et al.*, 1990] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artif. Intelligence*, 44(1-2):167–207, 1990.

[Kuželka *et al.*, 2015] O. Kuželka, J. Davis, and S. Schockaert. Encoding markov logic networks in possibilistic logic. In *Uncertainty in Artificial Intelligence, UAI*, 2015.

[Lang, 2001] J. Lang. Possibilistic logic: complexity and algorithms. In J. Kohlas and S. Moral, editors, *Algorithms for Uncertainty and Defeasible Reasoning*, volume 5 of *Handbook of Defeasible Reasoning and Uncertainty Management Systems (Gabbay D., Smets P. Eds.)*, pages 179–220. Kluwer Academic Publishers, 2001.

[Lehmann and Magidor, 1992] D. Lehmann and M. Magidor. What does a conditional knowledge base entail? *Artificial Intelligence*, 55(1):1–60, 1992.

[Lowd and Davis, 2014] D. Lowd and J. Davis. Improving markov network structure learning using decision trees. *Journal of Machine Learning Research*, 15:501–532, 2014.

[Pearl, 1990] J. Pearl. System Z: A natural ordering of defaults with tractable applications to nonmonotonic reasoning. In *3rd Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 121–135, 1990.

[Quinlan, 1993] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[Richardson and Domingos, 2006] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

[Serrurier and Prade, 2007] M. Serrurier and H. Prade. Introducing possibilistic logic in ILP for dealing with exceptions. *Artificial Intelligence*, 171(1617):939 – 950, 2007.

[Vapnik, 1995] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.