Learning Possibilistic Logic Theories from Default Rules (Abridged Version)

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". 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. *This short paper is an abridged version of [Kuželka* et al.*, 2016].*

1 Introduction

Structured information plays an increasingly important role in applications such as information extraction, question answering and robotics. 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. Trump 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.

2 Background: 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 $\mathit{inc}(\Theta)$ 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: $\Theta \vdash_{poss} \alpha$ iff $\Theta_{\overline{\mathit{inc}(\Theta)}} \models \alpha$. 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 certaintly levels in Θ [Lang, 2001].

3 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 $e = \alpha \mid\!\sim \beta$ by checking if h covers e, in the following sense.

Definition 1 (Covering). *A hypothesis $h \in \mathcal{H}$* covers *an example $e = \alpha \mid\!\sim \beta$ if $(h, \alpha) \vdash_{poss} \beta$.*

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

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

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

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

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

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* $\mathcal{S}$. This 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.

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

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 $\mathcal{S}^-$. Because arbitrary stratifications can be chosen, there is substantial freedom to ensure that negative examples are not covered[1]. Unfortunately, the problem of finding a separating 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.*

Another important parameter besides computational complexity is *sample complexity* which can be determined using the Vapnik-Chervonenkis (VC) dimension. Hence, we also determine the VC of the set of possible stratifications of a propositional theory. Let us write $\mathit{Strat}(\mathcal{T})$ for the set of all stratifications of a theory $\mathcal{T}$, and let $\mathit{Strat}^{(k)}(\mathcal{T})$ 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 $\mathit{Strat}^{(k)}(\mathcal{T})$.

Theorem 2. *Let* $\mathcal{T}$ *be a set of* n *propositional formulas. Then* $VC(\mathit{Strat}^{(k)}(\mathcal{T})) \leq n \log_2 k$.

The next theorem establishes a lower bound on the VC dimension of stratifications with at most k levels.

Theorem 3. *For every* k, n, $k \leq n$, *there is a propositional theory* $\mathcal{T}$ *consisting of* n *formulas such that*

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

4 Experiments with a Heuristic Algorithm

We implemented a heuristic algorithm, which combined structure learning and parameter learning, and we evaluated it in two different applications: learning of 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 thus compared our method to three traditional classifiers on two crowdsourced datasets of parallel rules. In the second experiment, approximating MAP inference, we did not restrict ourselves to parallel rules. In this case, only our method can guarantee that the predicted defaults will be consistent. This would also be the case if we did not ask the crowdsourcers only about "parallel" rules. The experimental results and details of the methodology are described in the full version of this paper [Kuželka *et al.*, 2016].

[1]Note that, as we show in [Kuželka *et al.*, 2016], to decide whether a separating stratification exists it is not sufficient to compute the Z-ranking because of the presence of negative examples.

5 Future Work

There are several important directions for future work. Although our implementation is capable of working with tens of thousands of defaults, it still does not scale to datasets of the sizes of knowledge bases such as FreeBase. Scalability is therefore an important issue. Also while possibilistic logic is a natural choice for representing the learned default rule theory, the framework of learning from default rules, which we introduced here, could as well work with other representations of default rules which might be more suitable for certain domains. On the applications side, it would be interesting to apply the method to learning from symtoms and diagnoses/treatments.

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

[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.

[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.

[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.*, 2016] Ondřej Kuželka, Jesse Davis, and Steven Schockaert. Learning Possibilistic Logic Theories from Default Rules. In *Proceedings of IJCAI*, 2016.

[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, 2001.