# INFORMATIK
## BERICHTE

**361 – 09/2011**


# Evolving Knowledge in Theory and Applications

**3rd Workshop on**
**Dynamics of Knowledge and Belief**
**(DKB 2011)**

**at the 34th Annual German Conference on Artifical Intelligence, KI-2011**
**Berlin, Germany, October 4, 2011**

**Proceedings**


**Christoph Beierle, Gabriele Kern-Isberner (Eds.)**




**Fakultät für Mathematik und Informatik**
**Postfach 940**
**D-58084 Hagen**

Christoph Beierle, Gabriele Kern-Isberner (Eds.)

# Evolving Knowledge in Theory and Applications

3rd Workshop on
Dynamics of Knowledge and Belief
(DKB-2011)

at the 34th Annual German Conference
on Artificial Intelligence, KI-2011
Berlin, Germany, October 4, 2011

Proceedings

# Workshop Organization

Special Interest Group on Knowledge Representation and Reasoning
of the Gesellschaft für Informatik

(GI-Fachgrupppe "Wissensrepräsentation und Schließen")

## Workshop Co-Chairs

| | |
|---|---|
| Gabriele Kern-Isberner | Technische Universität Dortmund, Germany |
| Christoph Beierle | FernUniversität in Hagen, Germany |

## Program Committee

| | |
|---|---|
| François Bry | LMU München, Germany |
| Jürgen Dix | TU Clausthal-Zellerfeld, Germany |
| Joachim Hertzberg | Universität Osnabrück, Germany |
| Pascal Hitzler | Wright State University, Dayton, Ohio, USA |
| Dieter Hutter | DFKI, Bremen, Germany |
| Haythem O. Ismail | German University in Cairo, Egypt |
| Radim Jirousek | UTIA, Academy of Sciences, Prague |
| Kristian Kersting | Fraunhofer IAIS, Universität Bonn, Germany |
| Michael Kohlhase | Jacobs University, Bremen, Germany |
| Markus Krötzsch | University of Oxford, UK |
| Gerhard Lakemeyer | RWTH Aachen, Germany |
| Claus Möbus | Carl von Ossietzky Universität, Oldenburg, Germany |
| Laurent Perrussel | University of Toulouse 1 Capitole, France |
| Odinaldo Rodrigues | King's College London, UK |
| Sebastian Rudolph | Karlsruher Institut für Technologie, Germany |
| Torsten Schaub | Universität Potsdam, Germany |
| Matthias Thimm | TU Dortmund, Germany |
| Hans Tompits | TU Wien, Austria |
| Gerhard Weiss | Maastricht University, The Netherlands |

## Additional Reviewer

Martin Günther

# Preface

In recent years, intelligent agents in the contexts of open environments and multi agent systems have become a leading paradigm in AI. Acting successfully in such environments that are uncertain, only partially accessible, and dynamic, requires sophisticated knowledge representation and reasoning techniques for the modelling of the epistemic state of the agent. In particular, in evolving environments, the agent must continuously react to new observations and to any unforeseen changes that occur. Its epistemic state must undergo corresponding changes to provide the agent with a suitable world view at any time. Thus, modern knowledge representation methods have to deal with the evolution of knowledge and belief, due to uncertain or incomplete information, or to changes in the environment.

This volume contains the contributions that were presented at the Workshop *Evolving Knowledge in Theory and Applications* on October 4, 2011, in Berlin, Germany, co-located with the 34th Annual German Conference on AI (KI-2011). This workshop was the 3rd Workshop on "Dynamics of Knowledge and Belief" (DKB-2011) organized by the Special Interest Group on Knowledge Representation and Reasoning of the Gesellschaft für Informatik (*GI-Fachgrupppe Wissensrepräsentation und Schließen*), following two previous workshops at KI-2007 in Osnabrück and at KI-2009 in Paderborn. The particular focus of the workshop was on any topics of knowledge representation and reasoning that address the epistemic modelling of agents in open environments, and in particular on processes concerning evolving knowledge and belief both in theory and in applications.

The workshop started with a session on modelling and reasoning in probabilistic approaches. In his paper *On Prototypical Indifference and Lifted Inference in Relational Probabilistic Conditional Logic*, Matthias Thimm investigates the complexity of probabilistic reasoning in a relational setting. Based on the notion of prototypical indifference he shows that lifted inference is no longer exponential in the number of domain elements when all predicates are unary, but is still infeasible for the general case.

Markov logic is a formalism generalising both first-order logic (for finite domains) and probabilistic graphical models. In the contribution *Knowledge Engineering with Markov Logic Networks: A Review*, Dominik Jain addresses knowledge engineering aspects with Markov logic. He describes the fundamental semantics of Markov logic networks, explains how simple modelling invariants can be represented, and discusses some fallacious modelling assumptions.

The third paper of this session deals with notions of probabilistic inconsistencies. In *Analyzing Inconsistencies in Probabilistic Conditional Knowledge Bases using Continuous Inconsistency Measures*, Matthias Thimm discusses the problem of analyzing and measuring inconsistencies in probabilistic conditional logic by investigating inconsistency measures that support the knowledge engineer in maintaining a consistent knowledge base. He develops continuous inconsistency measures assigning a numerical value to the severity of an inconsistency which can

be used for restoring consistency.

In the next session, three papers investigating aspects of relational probabilistic learning were presented. In their joint paper *Learning Scenarios under Relational Probabilistic Semantics and ME Reasoning*, Marc Finthammer and Nico Potyka present a learning scenario for relational learning which takes statistics on a population as well as uncertainty on individuals into account. Developed as an extension of a propositional maximum entropy framework, it is illustrated by various examples and compared to some popular statistical relational learning approaches like Markov Logic Networks and Bayesian Logic Programs.

In *Statistical Relational Learning in Dynamic Environments - An Agent-Based Approach to Traffic Navigation Using Bayesian Logic Networks*, Daan Apeldoorn uses Bayesian Logic Networks in a navigation application in a dynamic environment. Conditional probabilities are learned by an agent moving through a simulated traffic environment, and logical rules are added to determine the agent's behavior. The implementation of the agent is realized with the ProbCog Toolbox, an open-source software system for statistical relational learning.

Finally, the paper *On Efficient Algorithms for Minimal ME-Learning* by Nico Potyka studies probabilistic learning in the context of the principle of maximum entropy. This principle states that a set of probabilistic rules is best represented by the unique probability distribution satisfying all rules and possessing maximum entropy. While in previous work, an algebraic approach was used for realizing learning by inverting maximumg entropy inference, here learning is addressed with an approximative generate-and-test strategy.

We would like to thank all Program Committee members as well as the additional external reviewer for detailed and high-quality reviews for all submitted papers. Many thanks also to the organizers of KI-2011 for hosting the workshop at the KI-2011 conference. Finally, we would like to thank the Gesellschaft für Informatik, the TU Dortmund, and the FernUniversität in Hagen for supporting this workshop.

September 2011 Gabriele Kern-Isberner and Christoph Beierle

# Contents

# On Prototypical Indifference and Lifted Inference in Relational Probabilistic Conditional Logic

Matthias Thimm

Technische Universität Dortmund, Germany

**Abstract.** Semantics for formal models of probabilistic reasoning rely on probability functions that are defined on the interpretations of the underlying classical logic. When this underlying logic is of relational nature, i. e. a fragment of first-order logic, then the space needed for representing these probability functions explicitly is exponential in both the number of predicates and the number of domain elements. Consequently, probabilistic reasoning becomes a demanding task. Here, we investigate *lifted inference* in the context of explicit model representation with respect to an inference operator that satisfies *prototypical indifference*, i. e. an inference operator that is indifferent about individuals for which the same information is represented. As reasoning based on the principle of maximum entropy satisfies this property we exemplify our ideas by compactly characterizing the maximum entropy model of a probabilistic knowledge base in a relational probabilistic conditional logic. Our results show that lifted inference is no longer exponential in the number of domain elements when we restrict the language to unary predicates but is still infeasible for the general case.

## 1 Introduction

Applying probabilistic reasoning to relational representations of knowledge is a topic that has been mostly investigated within the fields of *statistical relational learning* and *probabilistic inductive logic programming* [3]. Those areas have put forth a variety of approaches that deal with combining traditional probabilistic models of knowledge like Bayes nets or Markov nets [10] with first-order logic, see e. g. Bayesian logic programs (BLPs) [3, Ch. 10] and Markov logic networks (MLNs) [3, Ch. 12]. Those frameworks employ knowledge-based model construction techniques [16] to reduce the problem of probabilistic reasoning in a relational context to probabilistic reasoning in a propositional context by appropriately grounding the parts of the knowledge base that are needed for answering a particular query.

In this paper we continue work on *relational probabilistic conditional logic* (RPCL) [6,15] which is a formalism for relational probabilistic knowledge representation that is apt for default reasoning as well. In RPCL uncertain knowledge is represented using probabilistic conditionals, i. e. *if-then* rules. Consider the following conditionals which represent both generic and specific rules of how

elephants like their keepers (the example is inspired by [2]):

$$r_1 =_{def} (likes(\mathsf{X},\mathsf{Y}) \mid elephant(\mathsf{X}) \wedge keeper(\mathsf{Y}))[0.6]$$
$$r_2 =_{def} (likes(\mathsf{X},\mathsf{fred}) \mid elephant(\mathsf{X}) \wedge keeper(\mathsf{fred}))[0.4]$$
$$r_3 =_{def} (likes(\mathsf{clyde},\mathsf{fred}) \mid elephant(\mathsf{clyde}) \wedge keeper(\mathsf{fred}))[0.7]$$

In many approaches to statistical relational learning such as BLPs relational rules are grounded, and the probability is attached to each instance. There, $r_1$ becomes the set $\{(likes(\mathsf{a},\mathsf{b}) \mid elephant(\mathsf{a}) \wedge keeper(\mathsf{b}))[0.6] \mid \mathsf{a},\mathsf{b} \in U\}$ where $U$ is some pool of constant symbols. As one can see, using naive grounding approaches renders the set of probabilistic conditionals from above inconsistent as there are instances of $r_1$ which contradict instances of $r_2$ and $r_3$. In [6] two novel semantics for relational probabilistic conditionals are introduced that avoid this problem. Further, by employing the principle of maximum entropy [9] one obtains a commonsense reasoning behavior [15].

Probabilistic reasoning in relational domains is, in general, a demanding task and there has been some efforts to speed up inference by exploiting structural equivalence in probabilistic knowledge [11]. This so-called *lifted inference* has been applied to e. g. parametrized belief networks and performed well in empirical experiments, cf. [13,8]. In this paper, we investigate *lifted inference* in RPCL. Our approach relies on the property of *prototypical indifference* which is satisfied by the maximum entropy approaches proposed in [6,15]. Basically, this property states that if a knowledge base $\mathcal{R}$ contains exactly the same information for constants $\mathsf{c}_1$ and $\mathsf{c}_2$ then reasoning with $\mathcal{R}$ is indifferent with respect to $\mathsf{c}_1$ and $\mathsf{c}_2$. Consequently, the maximum entropy models of [6,15] carry a lot of redundant information. We introduce *condensed probability functions* as a compact way to represent those probability functions. Condensed probability functions are defined on *reference worlds* which subsume a whole set of first-order interpretations that model the same situation modulo exchanging equivalent constants. Using reference worlds and condensed probability functions we rephrase the maximum entropy models of [6,15] in a computationally feasible way.

The rest of this paper is organized as follows. In Section 2 we briefly review the semantical and inferential approaches of [6,15]. In Section 3 we introduce condensed probability functions as a compact way to represent prototypically uniform probability functions. Afterwards, we propose our approach to lifted inference in Section 4 and analyze its advantages in Section 5. In Section 6 we briefly discuss the issue of extending our approach to non-unary languages. In Section 7 we review related work and in Section 8 we conclude. All proofs of technical results can be found in an online appendix[1].

## 2 Relational Probabilistic Conditional Logic and Inductive Reasoning

In the following, we give a brief overview on the syntax of *relational probabilistic conditional logic* (RPCL) and *averaging* and *aggregating semantics*, see [6] for a

[1] http://ls1-www.cs.tu-dortmund.de/~thimm/misc/thimm_lifted_dkb11_proofs.pdf

discussion. We consider only a fragment of a first-order language, so let $\Sigma$ be a first-order signature consisting of a finite set of predicate symbols and without functions of arity greater than zero. We also assume that $\Sigma$ contains some fixed and finite set of constant symbols $U_\Sigma$, i. e. functions of arity zero. An *atom* is a predicate together with some terms (variables or constant symbols), e. g., if $a/3$ is a predicate of arity three, $\mathsf{a}, \mathsf{b} \in U_\Sigma$ and $\mathsf{X}$ is a variable then $a(\mathsf{a}, \mathsf{X}, \mathsf{b})$ is an atom. Let $\mathcal{L}_\Sigma$ be the corresponding first-order language over the signature $\Sigma$ that is generated in the usual way using negation, conjunction, and disjunction, but without quantifiers. If appropriate we abbreviate conjunctions $\phi \wedge \psi$ by $\phi\psi$ and negation $\neg\psi$ by $\overline{\psi}$. We denote constants with a beginning lowercase, variables with a beginning uppercase letter, and vectors of these with $\vec{\mathsf{a}}$ resp. $\vec{\mathsf{X}}$.

The central notion of RPCL is the probabilistic conditional, see also [12].

**Definition 1.** *Let $\psi, \phi \in \mathcal{L}_\Sigma$ be some formulas and $d \in [0, 1]$. Then $(\psi \mid \phi)[d]$ is called a* probabilistic conditional.

A probabilistic conditional $(\psi \mid \phi)[d]$ is meant to represent the uncertain rule "if $\phi$ then $\psi$ with probability $d$". If $\phi$ is tautological, i. e., if $\phi \equiv \top$, we abbreviate $(\psi \mid \phi)[d]$ by $(\psi)[d]$ and call $(\psi)[d]$ a *probabilistic fact*. A *knowledge base* $\mathcal{R}$ is a finite set of probabilistic conditionals.

For a formula $\phi \in \mathcal{L}_\Sigma$ let $\mathsf{Const}(\phi) \subseteq U_\Sigma$ denote the set of constant symbols appearing in $\psi$ and let $\mathsf{Var}(\psi)$ denote the set of variable symbols appearing in $\psi$. The operators $\mathsf{Const}(\cdot)$ and $\mathsf{Var}(\cdot)$ are extended to probabilistic conditionals and knowledge bases in the usual way. Let now $x$ be either a formula, a probabilistic conditional, or a knowledge base. If $\mathsf{Var}(x) = \emptyset$ then $x$ is called *ground*. Furthermore, let $\mathsf{gnd}_\Sigma(x)$ denote the set of ground instances of $x$ with respect to the constant symbols in $U_\Sigma$. For example, if $U_\Sigma = \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3\}$ and $a(X) \in \mathcal{L}_\Sigma$ then $\mathsf{gnd}_\Sigma(a(\mathsf{X})) = \{a(\mathsf{c}_1), a(\mathsf{c}_2), a(\mathsf{c}_3)\}$. Note that for ground $x$ we have $\mathsf{gnd}_\Sigma(x) = \{x\}$.

In order to interpret the classical formulas within conditionals we use *Herbrand interpretations*, i. e. sets of ground atoms of $\Sigma$. Let $\Omega(\Sigma)$ denote the set of all Herbrand interpretations for the signature $\Sigma$. If $\phi \in \mathcal{L}_\Sigma$ is ground then $\omega \in \Omega(\Sigma)$ *satisfies* $\phi$, denoted by $\omega \models^{\mathrm{F}} \phi$, by the usual definition. Note that every $\omega \in \Omega(\Sigma)$ is finite and $\Omega(\Sigma)$ is finite as well as $\Sigma$ contains only finitely many predicate and constant symbols. Semantics are given to relational probabilistic conditionals by means of probability functions $P : \Omega(\Sigma) \to [0, 1]$, so let $\mathcal{P}^{\mathrm{F}}(\Sigma)$ denote the set of all these probability functions. A probability function $P \in \mathcal{P}^{\mathrm{F}}(\Sigma)$ is extended to ground formulas $\phi \in \mathcal{L}_\Sigma$ via

$$P(\phi) =_{def} \sum_{\omega \in \Omega(\Sigma),\ \omega \models^{\mathrm{F}} \phi} P(\omega) \quad . \tag{1}$$

The approach of *averaging semantics* interprets a probabilistic conditional $r = (\psi \mid \phi)[d]$ with variables by imposing that the average conditional probability of the instances of $(\psi \mid \phi)[d]$ matches $d$. For a probability function $P \in \mathcal{P}^{\mathrm{F}}(\Sigma)$ we abbreviate with $\mathsf{gnd}^P_\Sigma(r) =_{def} \{(\phi' \mid \psi')[d] \in \mathsf{gnd}_\Sigma(r) \mid P(\psi') > 0\}$ the set of ground instances of $r$ for which the premise has a non-zero probability in $P$.

Then $P$ $\varnothing$-satisfies $r = (\psi \mid \phi)[d]$ ($P \models^{pr}_{\varnothing} r$) iff $|\mathsf{gnd}^P_\Sigma(r)| > 0$ and

$$\frac{\sum_{(\psi' \mid \phi')[d] \in \mathsf{gnd}^P_\Sigma((\psi \mid \phi)[d])} P(\psi' \mid \phi')}{|\mathsf{gnd}^P_\Sigma((\psi \mid \phi)[d])|} = d \quad . \tag{2}$$

The interpretation behind the above equation is that a probability function $P$ $\varnothing$-satisfies a probabilistic conditional $(\psi \mid \phi)[d]$ if the average of probabilities of the individual instances of $(\psi \mid \phi)[d]$ is $d$. By considering only those ground instances where the premise has probability greater zero we average only over the probabilities of ground instances that are relevant for the open conditional.

*Example 1.* Consider the probabilistic conditional $r = (b(\mathsf{X}) \mid a(\mathsf{X}))[0.7]$ and $U_\Sigma = \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3\}$. Let $P$ be a probability function with $P(a(\mathsf{c}_1)) > 0$, $P(a(\mathsf{c}_2)) > 0$, and $P(a(\mathsf{c}_3)) > 0$. If e. g. $P(b(\mathsf{c}_1) \mid a(\mathsf{c}_1)) = 0.9$, $P(b(\mathsf{c}_2) \mid a(\mathsf{c}_2)) = P(b(\mathsf{c}_3) \mid a(\mathsf{c}_3)) = 0.6$ then $P \models^{pr}_{\varnothing} r$.

The similar approach of *aggregating semantics* is defined as follows[2]. A probability function $P$ $\odot$-satisfies a probabilistic conditional $r = (\psi \mid \phi)[d]$ ($P \models^{pr}_{\odot} (\psi \mid \phi)[d]$) iff $\sum_{(\psi' \mid \phi')[d] \in \mathsf{gnd}^P_\Sigma((\psi \mid \phi)[d])} P(\phi') > 0$ and

$$\frac{\sum_{(\psi' \mid \phi')[d] \in \mathsf{gnd}_\Sigma((\psi \mid \phi)[d])} P(\psi'\phi')}{\sum_{(\psi' \mid \phi')[d] \in \mathsf{gnd}_\Sigma((\psi \mid \phi)[d])} P(\phi')} = d \quad .$$

For a knowledge base $\mathcal{R}$ it holds that $P \models^{pr}_{\varnothing} \mathcal{R}$ ($P \models^{pr}_{\odot} \mathcal{R}$) iff $P \models^{pr}_{\varnothing} r$ ($P \models^{pr}_{\odot} r$) for all $r \in \mathcal{R}$. A knowledge base $\mathcal{R}$ is $\varnothing$-consistent ($\odot$-consistent) if there is a probability function $P$ with $P \models^{pr}_{\varnothing} \mathcal{R}$ ($P \models^{pr}_{\odot} \mathcal{R}$). Both averaging and aggregating semantics are extensions of the standard semantics for propositional probabilistic conditional logic. More precisely, if $r = (\psi \mid \phi)[d]$ is a ground probabilistic conditional, i. e. $\mathsf{Var}(r) = \emptyset$, then $P \models^{pr}_{\varnothing} r$ iff $P \models^{pr}_{\odot} r$ iff $P(\phi) > 0$ and $P(\psi \mid \phi) = {}^{P(\psi\phi)}/_{P(\phi)} = d$. However, the semantics are quite different in general, see [14] for a discussion.

In the following, let $\circ \in \{\varnothing, \odot\}$ be one of the semantics presented above. One can define a model-based inductive reasoning operator $\mathcal{I}_\circ$—which maps a knowledge base $\mathcal{R}$ onto a "suitable" probability function $\mathcal{I}_\circ(\mathcal{R})$ with $\mathcal{I}_\circ(\mathcal{R}) \models^{pr}_\circ \mathcal{R}$—as follows, cf. [9]. Let the *entropy* $H(P)$ of a probability function $P \in \mathcal{P}^{\mathrm{F}}(\Sigma)$ be defined via $H(P) = -\sum_{\omega \in \Omega(\Sigma)} P(\omega) \mathsf{ld}\, P(\omega)$.[3] The entropy measures the amount of indeterminateness of a probability function $P$. By selecting a model of a knowledge base $\mathcal{R}$ that has maximal entropy one gets a probability function that both satisfies all conditionals in $\mathcal{R}$ and adds as less additional information (in the information-theoretic sense) as possible [9].

**Definition 2.** *Let $\mathcal{R}$ be $\circ$-consistent. Then the* maximum entropy model $\mathcal{I}_\circ(\mathcal{R})$ *of $\mathcal{R}$ is defined via*

$$\mathcal{I}_\circ(\mathcal{R}) = \arg \max_{P \models^{pr}_\circ \mathcal{R}} H(P) \quad . \tag{3}$$

[2] For a justification for the aggregating semantics see [6,14].

[3] $\mathsf{ld}\, x$ is the binary logarithm of $x$ with $0\, \mathsf{ld}\, 0 = 0$.

Note that $\mathcal{I}_{\varnothing}(\mathcal{R})$ has not yet proven to be well-defined in general, see [14] for a discussion. But—as this issue is not the topic of the current work—we assume in the following that (3) is always well-defined.

Inference via $\mathcal{I}_{\circ}$ satisfies a series of rationality postulates such as the System P properties [15] and several properties for relational probabilistic reasoning [14]. One of this properties is *prototypical indifference* which can be exploited for our purpose of lifted inference. In order to state this property we need some further notation. If $x$ is either a formula, a probabilistic conditional, or a knowledge base and $\mathsf{c}_1, \mathsf{c}_2 \in U_\Sigma$ then $x[\mathsf{c}_1 \leftrightarrow \mathsf{c}_2]$ is the same as $x$ except that every occurrence of $\mathsf{c}_1$ is replaced with $\mathsf{c}_2$ and vice versa.

**Definition 3.** *Let $\mathcal{R}$ be a knowledge base. The constants $\mathsf{c}_1, \mathsf{c}_2 \in U_\Sigma$ are* $\mathcal{R}$-equivalent *($\mathsf{c}_1 \equiv_{\mathcal{R}} \mathsf{c}_2$) iff $\mathcal{R} = \mathcal{R}[\mathsf{c}_1 \leftrightarrow \mathsf{c}_2]$.*

Observe that $\equiv_{\mathcal{R}}$ is indeed an equivalence relation. Two $\mathcal{R}$-equivalent constants $\mathsf{c}_1$ and $\mathsf{c}_2$ are indistinguishable with respect to knowledge base $\mathcal{R}$. That is, $\mathcal{R}$ models exactly the same knowledge on both $\mathsf{c}_1$ and $\mathsf{c}_2$. Also note that every two $\mathsf{c}_1, \mathsf{c}_2 \in U_\Sigma$ with $\mathsf{c}_1, \mathsf{c}_2 \notin \mathsf{Const}(\mathcal{R})$ are $\mathcal{R}$-equivalent.

**Definition 4.** *A set $S = \{\mathsf{c}' \mid \mathsf{c}' \equiv_{\mathcal{R}} \mathsf{c}\} \subseteq U_\Sigma$ for $\mathsf{c} \in U_\Sigma$ is called* $\mathcal{R}$-equivalence class *and $\mathfrak{S}(\mathcal{R})$ is the set of all $\mathcal{R}$-equivalence classes.*

In [6] it has been shown that $\mathcal{I}_{\circ}$ satisfies the following property of *prototypical indifference*.

**Theorem 1 (Prototypical indifference).** *If $\mathcal{R}$ is $\circ$-consistent then $\mathsf{c}_1 \equiv_{\mathcal{R}} \mathsf{c}_2$ implies $\mathcal{I}_{\circ}(\mathcal{R})(\psi) = \mathcal{I}_{\circ}(\mathcal{R})(\psi[\mathsf{c}_1 \leftrightarrow \mathsf{c}_2])$ for every ground formula $\psi$.*

The above theorem implies that the probability function $\mathcal{I}_{\circ}(\mathcal{R})$ carries a lot of redundant information. Consider the following example.

*Example 2.* Let $U_\Sigma =_{def} \{\mathsf{tweety}, \mathsf{huey}, \mathsf{dewey}, \mathsf{louie}\}$ be a set of constant symbols and let $\mathcal{R}_{\text{birds}} =_{def} \{(\mathit{flies}(\mathsf{X}))[0.8], (\mathit{flies}(\mathsf{tweety}))[0.3]\}$ be a knowledge base stating that 80 % of all birds fly and that Tweety flies only up to a degree of belief of 0.3. Consider now the probability function $P^* = \mathcal{I}_{\circ}(\mathcal{R}_{\text{birds}})$ which is defined on the set of Herbrand interpretations $\Omega(\Sigma) = \{\omega_0, \ldots, \omega_{15}\}$ with e. g.

$$\omega_5 =_{def} \{\mathit{flies}(\mathsf{tweety}), \mathit{flies}(\mathsf{huey})\} \qquad \omega_6 =_{def} \{\mathit{flies}(\mathsf{tweety}), \mathit{flies}(\mathsf{dewey})\}$$
$$\omega_7 =_{def} \{\mathit{flies}(\mathsf{tweety}), \mathit{flies}(\mathsf{louie})\} \qquad \omega_8 =_{def} \{\mathit{flies}(\mathsf{huey}), \mathit{flies}(\mathsf{dewey})\}$$
$$\omega_9 =_{def} \{\mathit{flies}(\mathsf{huey}), \mathit{flies}(\mathsf{louie})\} \qquad \omega_{10} =_{def} \{\mathit{flies}(\mathsf{dewey}), \mathit{flies}(\mathsf{louie})\}$$

The $\mathcal{R}$-equivalence classes $\mathfrak{S}(\mathcal{R}) = \{S_1, S_2\}$ of $\mathcal{R}$ are given by $S_1 = \{\mathsf{tweety}\}$ and $S_2 = \{\mathsf{huey}, \mathsf{dewey}, \mathsf{louie}\}$ and due to Theorem 1 it follows that e. g. $P^*(\psi) = P^*(\psi[\mathsf{huey} \leftrightarrow \mathsf{dewey}])$ for every ground sentence $\psi$. In particular, as every $\omega \in \Omega(\Sigma)$ can be understood as a ground conjunction we obtain $P^*(\omega_5) = P^*(\omega_6) = P^*(\omega_7)$. Therefore, it suffices to represent $P^*$ by only eight Herbrand interpretations as the other eight contain only redundant information.

For the rest of this paper we elaborate on the idea suggested in the above example.

## 3 Condensed Probability Functions

In the previous section the notion of $\mathcal{R}$-equivalence has been introduced as a relation among constant symbols, cf. Definition 3. We can generalize this relation to be applicable on Herbrand interpretations as follows. Let $\mathfrak{S}(\mathcal{R}) = \{S_1, \ldots, S_n\}$.

**Definition 5.** *Let $\omega_1, \omega_2 \in \Omega(\Sigma)$. Then $\omega_1$ and $\omega_2$ are* $\mathcal{R}$-equivalent*, denoted by $\omega_1 \equiv_{\mathcal{R}} \omega_2$, if there is some $G \in \mathbb{N}$ and a set $T = \{(\mathsf{c}_1^1, \mathsf{c}_2^1), \ldots, (\mathsf{c}_1^G, \mathsf{c}_2^G)\} \subseteq S_1 \times S_1 \cup \ldots \cup S_n \times S_n$ such that $\omega_1 = \omega_2[\mathsf{c}_1^1 \leftrightarrow \mathsf{c}_2^1] \ldots [\mathsf{c}_1^G \leftrightarrow \mathsf{c}_2^G]$.*

Basically, $\omega_1$ and $\omega_2$ are $\mathcal{R}$-equivalent if we can permute elements within each $\mathcal{R}$-equivalence class such that $\omega_2$ becomes $\omega_1$, e. g. in Example 2 we have $\omega_5 \equiv_{\mathcal{R}_{\text{birds}}} \omega_6 \equiv_{\mathcal{R}_{\text{birds}}} \omega_7$. It is also easy to see that $\equiv_{\mathcal{R}}$ is an equivalence relation and, therefore, both the $\mathcal{R}$-equivalence class $[\omega] =_{def} \{\omega' \in \Omega(\Sigma) \mid \omega \equiv_{\mathcal{R}} \omega'\}$ and the quotient set $\Omega(\Sigma)/_{\equiv_{\mathcal{R}}} =_{def} \{[\omega] \mid \omega \in \Omega(\Sigma)\}$ are well-defined.

**Proposition 1.** *If $\omega_1 \equiv_{\mathcal{R}} \omega_2$ then $\mathcal{I}_\circ(\omega_1) = \mathcal{I}_\circ(\omega_2)$.*

The above proposition states that the probability function $\mathcal{I}_\circ$ carries a lot of redundant information stemming from the $\mathcal{R}$-equivalence of certain $\omega \in \Omega(\Sigma)$. In the following, we exploit this observation by using $\Omega(\Sigma)/_{\equiv_{\mathcal{R}}}$ instead of $\Omega(\Sigma)$ for redefining $\mathcal{I}_\circ$. To do so, we go on by developing a method that enumerates the elements of $\Omega(\Sigma)/_{\equiv_{\mathcal{R}}}$ in an effective way.

For the rest of this section we restrain our attention to signatures containing only unary predicates. Therefore, let $Pred =_{def} \{p_1, \ldots, p_P\}$ be the set of unary predicates of $\Sigma$. We discuss the issue of generalizing our approach in Section 6.

**Definition 6.** *A* truth configuration *$t$ for $Pred$ is an expression $t =_{def} \dot{p}_1 \ldots \dot{p}_P$ with $\dot{p}_i \in \{p_i, \overline{p}_i\}$ for $i = 1, \ldots, P$. Let $\Theta$ denote the set of all truth configurations.*

A truth configuration is meant to characterize the state of a constant $\mathsf{c}$ in some interpretation as it enumerates which predicates apply for $\mathsf{c}$ and which do not. For a constant $\mathsf{c}$ and a truth configuration $t = \dot{p}_1 \ldots \dot{p}_P$ define $t^\wedge(\mathsf{c}) =_{def} \dot{p}_1(\mathsf{c}) \wedge \ldots \wedge \dot{p}_P(\mathsf{c})$. Furthermore, for a ground sentence $\phi$ and constants $\mathsf{c}_1, \ldots, \mathsf{c}_n$ let

$$\Theta(\phi, \mathsf{c}_1) =_{def} \{t \in \Theta \mid t^\wedge(\mathsf{c}_1) \wedge \phi \not\models^{\mathrm{F}} \bot\}$$
$$\Theta(\phi, \mathsf{c}_1, \ldots, \mathsf{c}_n) =_{def} \Theta(\phi, \mathsf{c}_1) \times \ldots \times \Theta(\phi, \mathsf{c}_n) \quad .$$

The set $\Theta(\phi, \mathsf{c}_1)$ contains all those truth configurations $t$ for a constant $\mathsf{c}_1$ that are compatible with some sentence $\phi$. The set $\Theta(\phi, \mathsf{c}_1, \ldots, \mathsf{c}_n)$ extends this notion to tuples of constants.

*Example 3.* Let $Pred =_{def} \{p_1/1, p_2/1\}$ and let $\psi =_{def} p_1(\mathsf{c}) \wedge (p_2(\mathsf{c}) \vee p_2(\mathsf{d}))$. Then it holds that $\Theta(\phi, \mathsf{c}) = \{p_1 p_2, p_1 \overline{p}_2\}$.

**Definition 7.** *An* instance assignment *$I$ is a function $I : \mathfrak{S}(\mathcal{R}) \to \mathbb{N}_0$ with $I(S_i) \leq |S_i|$ for all $i = 1, \ldots, n$. Let $\mathfrak{I}$ denote the set of all instance assignments.*

An instance assignment $I$ assigns to each $\mathcal{R}$-equivalence class the number of constants that are part of the current instance, see below.

**Definition 8.** *A* reference world $\hat{\omega}$ *is a function* $\hat{\omega} : \Theta \to \mathfrak{I}$ *that satisfies*

$$\sum_{t \in \Theta} \hat{\omega}(t)(S_i) = |S_i| \quad (\textit{for all } i = 1, \ldots, n) \quad . \tag{4}$$

*Let* $\hat{\Omega}$ *be the set of all reference worlds.*

Basically, a reference world is a function that maps a truth configuration to the number of constants of each $\mathcal{R}$-equivalence class that satisfy this truth configuration. As we show later, a reference world is a compact representation of $[\omega]$ for some $\omega \in \Omega(\Sigma)$.

*Example 4.* We continue Example 2. The set of truth configurations $\Theta = \{t_1, t_2\}$ with respect to $\Sigma$ and $\mathcal{R}_{\text{birds}}$ is given via $t_1 = \textit{flies}$ and $t_2 = \overline{\textit{flies}}$. Consider $I, I' \in \mathfrak{I}$ with $I(S_1) = 0$, $I(S_2) = 2$, $I'(S_1) = 1$, $I'(S_2) = 1$ and $\hat{\omega} \in \hat{\Omega}$ with $\hat{\omega}(t_1) = I$ and $\hat{\omega}(t_2) = I'$. The intuitive description of $\hat{\omega}$ is that $\hat{\omega}$ represents a state where the one element of $S_1$ does not fly and two elements of $S_2$ do fly.

In the following we show that $\hat{\Omega}$ is indeed a characterization of the quotient set $\Omega(\Sigma)/_{\equiv_{\mathcal{R}}}$. For that, consider the following definition.

**Definition 9.** *The* equivalence mapping $\kappa$ *is the function* $\kappa : \Omega(\Sigma) \to \hat{\Omega}$ *defined as* $\kappa(\omega) =_{def} \hat{\omega}$ *with* $\hat{\omega}(\dot{p}_1 \ldots \dot{p}_P)(S_i) =_{def} |\{\mathsf{c} \in S_i \mid \omega \models^F \dot{p}_1(\mathsf{c}) \wedge \ldots \wedge \dot{p}_P(\mathsf{c})\}|$ *for every* $\dot{p}_1 \ldots \dot{p}_P \in \Theta$ *and* $i = 1, \ldots, n$.

The function $\kappa$ maps a $\omega \in \Omega(\Sigma)$ onto a reference world $\hat{\omega} \in \hat{\Omega}$ with the intended meaning that $\kappa(\omega)$ is the (unique) reference world that represents $\omega$. It holds that $\kappa(\omega) = \hat{\omega}$ whenever $\hat{\omega}$ assigns the same number of elements of an $\mathcal{R}$-equivalence class $S_i$ to some truth configuration $t$ as $\omega$ contains specific instances of this truth configuration for elements in $S_i$. Also note that $\kappa$ is surjective.

Let the *span number* $\rho_{\hat{\omega}}$ of a reference world $\hat{\omega} \in \hat{\Omega}$ be defined as[4]

$$\rho_{\hat{\omega}} =_{def} \prod_{i=1}^{n} \binom{|S_i|}{\hat{\omega}(t_1)(S_i), \ldots, \hat{\omega}(t_T)(S_i)}$$

with $\Theta = \{t_1, \ldots, t_T\}$. Note that $\rho_{\hat{\omega}}$ is well-defined as $\hat{\omega}(t_1)(S_i) + \ldots + \hat{\omega}(t_T)(S_i) = |S_i|$ for every $\hat{\omega}$. The span number of $\hat{\omega}$ is exactly the number of Herbrand interpretations that are subsumed by $\hat{\omega}$.

**Proposition 2.** *It holds that* $|\kappa^{-1}(\hat{\omega})| = \rho_{\hat{\omega}}$ *for every* $\hat{\omega} \in \hat{\Omega}$.

The following proposition states that $\hat{\Omega}$ indeed characterizes $\Omega(\Sigma)/_{\equiv_{\mathcal{R}}}$.

**Proposition 3.** *The function* $\iota : \Omega(\Sigma)/_{\equiv_{\mathcal{R}}} \to \hat{\Omega}$ *with* $\iota([\omega]) = \kappa(\omega)$ *is a bijection.*

After having established the equivalence of $\Omega(\Sigma)/_{\equiv_{\mathcal{R}}}$ and $\hat{\Omega}$ we now turn to the issue of representing $\mathcal{I}_\circ$ on the basis of $\hat{\Omega}$.

[4] $\binom{n}{k_1, \ldots, k_r} =_{def} \frac{n!}{k_1! \cdots k_r!}$ is the *multinomial coefficient* indexed by $n$ and $k_1, \ldots, k_r$ with $n = k_1 + \ldots + k_r$ and $\binom{n}{k_1, \ldots, k_r} =_{def} 0$ if any $k_i < 0$ for $i = 1, \ldots, n$.

**Definition 10.** *A probability function $P : \Omega(\Sigma) \to [0,1]$ is called* prototypically uniform *wrt. $\mathcal{R}$ iff for $\omega_1, \omega_2 \in \Omega(\Sigma)$ with $\omega_1 \equiv_{\mathcal{R}} \omega_2$ it follows that $P(\omega_1) = P(\omega_2)$.*

Note that $\mathcal{I}_\circ(\mathcal{R})$ is prototypically uniform wrt. $\mathcal{R}$. Prototypically uniform probability functions can be be concisely represented as follows.

**Definition 11.** *Let $P$ be a probability function $P : \Omega(\Sigma) \to [0,1]$ that is prototypically uniform wrt. $\mathcal{R}$. Then the* condensed probability function $\hat{P}$ *for $P$ is the probability function $\hat{P} : \hat{\Omega} \to [0,1]$ defined via $\hat{P}(\hat{\omega}) =_{def} P(\omega)$ for some $\omega$ with $\kappa(\omega) = \hat{\omega}$ and for all $\hat{\omega} \in \hat{\Omega}$. Let $\hat{\mathcal{P}}$ denote the set of all condensed probability functions.*

As $\kappa(\omega_1) = \kappa(\omega_2)$ implies $P(\omega_1) = P(\omega_2)$ for prototypically uniform $P$ the function $\hat{P}$ is well-defined. It also holds that the mapping between prototypically uniform probability functions and condensed probability functions is bijective.

**Proposition 4.** *Let $P_1, P_2$ be prototypically uniform probability functions wrt. $\mathcal{R}$. It holds that $P_1 = P_2$ iff $\hat{P}_1 = \hat{P}_2$.*

For a prototypically uniform probability function $P$, its condensed probability function $\hat{P}$, and a ground sentence $\psi$ it follows directly by definition that

$$\hat{P}(\psi) =_{def} P(\psi) = \sum_{\omega \in \Omega(\Sigma),\ \omega \models^{\mathrm{F}} \psi} \hat{P}(\kappa(\omega)) \quad . \tag{5}$$

As one can see, one can determine the probability of any ground sentence using $\hat{P}$ instead of $P$. However, the sum in the above equation still considers every $\omega \in \Omega(\Sigma)$. In the next section we consider the question of how to determine the probability of $\psi$ *without* considering $\Omega(\Sigma)$ but only $\hat{\Omega}$ instead.

## 4 Lifted Inference

Looking closer at Equation (5) one can see that the probability of a $\hat{\omega} \in \hat{\Omega}$ may occur more than once within the sum as for different $\omega, \omega' \in \Omega(\Sigma)$ with $\omega \models^{\mathrm{F}} \psi$ and $\omega' \models^{\mathrm{F}} \psi$ it may hold that $\kappa(\omega) = \kappa(\omega')$. Therefore, (5) can be rewritten to

$$\hat{P}(\psi) = \sum_{\hat{\omega} \in \hat{\Omega}} \Lambda(\hat{\omega}, \psi) \hat{P}(\hat{\omega}) \quad . \tag{6}$$

with $\Lambda(\hat{\omega}, \psi) = |\{\omega \in \Omega(\Sigma) \mid \kappa(\omega) = \hat{\omega} \wedge \omega \models^{\mathrm{F}} \psi\}| \in \mathbb{N}_0$, i. e., $\Lambda(\hat{\omega}, \psi)$ is the number of $\omega \in \Omega(\Sigma)$ in (5) that satisfy $\psi$ and are mapped by $\kappa$ to $\hat{\omega}$. Note, however, that determining $\Lambda(\hat{\omega}, \psi)$ by its definition above still requires considering all $\omega \in \Omega(\Sigma)$. By exploiting combinatorial patterns within the structure of $\Omega(\Sigma)$ we can avoid considering $\Omega(\Sigma)$ as a whole and characterize $\Lambda(\hat{\omega}, \psi)$ as follows.

**Proposition 5.** *Let $\psi$ be a conjunction of ground literals, let $\mathsf{Const}(\psi) = \{\mathsf{c}_1, \ldots, \mathsf{c}_m\}$, and let $\Theta = \{t_1, \ldots, t_T\}$. Then*

$$\Lambda(\hat{\omega}, \psi) = \sum_{(t'_1,\ldots,t'_m) \in \Theta(\psi, \mathsf{c}_1, \ldots, \mathsf{c}_m)} \prod_{i=1}^{n} \binom{|S_i \setminus \mathsf{Const}(\phi)|}{\alpha_i^{t_1}(t'_1, \ldots, t'_m), \ldots, \alpha_i^{t_T}(t'_1, \ldots, t'_m)}$$

*with $\alpha_i^t(t'_1, \ldots, t'_m) =_{def} \hat{\omega}(t)(S_i) - |\{k \mid t'_k = t \wedge \mathsf{c}_k \in S_i\}|$.*

Note that there is no more reference to $\Omega(\Sigma)$ in the above characterization of $\Lambda(\hat{\omega}, \psi)$.

In order to determine $\hat{P}(\psi)$ for an arbitrary ground sentence $\psi$ remember that $\psi$ can be rewritten to be in disjunctive normal form. Assume $\psi$ to be in disjunctive normal form and let $c(\psi)$ denote the set of conjuncts of $\psi$. Then we can write

$$\hat{P}(\psi) = \sum_{\psi' \in c(\psi)} \hat{P}(\psi') - \sum_{(\psi', \psi'') \in c(\psi)^2,\ \psi' \neq \psi''} \hat{P}(\psi' \wedge \psi'') \quad .$$

As for $\hat{P}$, for every $\psi', \psi'' \in c(\psi)$ the terms $\hat{P}(\psi')$ and $\hat{P}(\psi' \wedge \psi'')$ are well-defined by Equation (6) and Proposition 5.

So far, we have shown how that $\hat{P^*}$ compactly represents $P^* = \mathcal{I}_\circ(\mathcal{R})$ and that $\hat{P^*}$ can be used for reasoning just as $P^*$. Nonetheless, in order to determine $\hat{P^*}$ one needs to compute $P^*$ first using Equation (3). In the following, we show that we can modify (3) in a straightforward fashion to determine $\hat{P^*}$ directly. Note, although the approach of condensed probability distributions is applicable to any inductive inference mechanism that obeys *prototypical indifference* we restrain our attention to $\mathcal{I}_\circ$.

For a condensed probability function $\hat{P}$ we define the *entropy* $H(\hat{P})$ of $\hat{P}$ to be the entropy of $P$, i. e. $H(\hat{P}) =_{def} H(P)$, which is equivalent to

$$H(\hat{P}) = -\sum_{\hat{\omega} \in \hat{\Omega}} \rho_{\hat{\omega}} \hat{P}(\hat{\omega}) \mathsf{ld}\, \hat{P}(\hat{\omega})$$

and thus can be determined by just considering $\hat{\Omega}$.

**Proposition 6.** *Let $\mathcal{S}$ be a set of prototypical uniform probability functions wrt. $\mathcal{R}$ and*

$$\hat{\mathcal{S}} =_{def} \{\hat{P} \mid P \in \mathcal{S}\} \quad .$$

*If the probability function $P_1 = \arg\max_{P \in \mathcal{S}} H(P)$ is uniquely determined so is $\hat{Q} = \arg\max_{\hat{P} \in \hat{\mathcal{S}}} H(\hat{P})$ and it holds that $\hat{Q} = \hat{P}_1$.*

**Proposition 7.** *Let $\mathcal{S} =_{def} \{P \mid P \models^{pr}_\circ \mathcal{R}\}$ and let $\mathcal{S}' \subseteq \mathcal{S}$ be its subset of prototypical uniform probability functions with respect to $\mathcal{R}$. If $\arg\max_{P \in \mathcal{S}} H(P)$ is uniquely determined then it holds that*

$$\arg\max_{P \in \mathcal{S}'} H(P) = \arg\max_{P \in \mathcal{S}} H(P) \quad .$$

The implications of the above two propositions are as follows. Instead of determining first $P^* = \mathcal{I}_\circ(\mathcal{R})$ via (3) and then determining $\hat{P^*}$ we can directly determine $\hat{P^*}$ by rewriting (3) to

$$\widehat{\mathcal{I}_\circ(\mathcal{R})} = \arg\max_{\hat{P} \in \hat{\mathcal{P}} \text{ and } \hat{P} \models^{pr}_\varnothing \mathcal{R}} H(P) \tag{7}$$

Note that $\hat{P} \models^{pr}_\circ \mathcal{R}$ can be checked directly for $\hat{P}$ by employing Equation (6) and Proposition 5.

## 5 Analysis

We now analyze the computational benefits of using $\hat{P}^*$ instead of $P^* = \mathcal{I}_\circ(\mathcal{R})$. In particular, we are interested in the question how the cardinality of $\hat{\Omega}$ compares to the cardinality of $\Omega(\Sigma)$ with respect to the number of constants $|U_\Sigma|$ considered. It is easy to see that $|\Omega(\Sigma)| = 2^{|U_\Sigma||Pred|}$ (remember that *Pred* is the set of (unary) predicates in $\Sigma$) and therefore the space needed to represent $P^*$ is exponential in both $|U_\Sigma|$ and $|Pred|$. We do not expect to avoid an exponential blow-up in the number of predicates in *Pred* but we show that $|\hat{\Omega}|$ is not exponential in $|U_\Sigma|$ any more. Remember that each $\hat{\omega} \in \hat{\Omega}$ satisfies

$$\sum_{t \in \Theta} \hat{\omega}(t)(S_i) = |S_i| \quad \text{(for all } i = 1, \ldots, n) \quad .$$

This means, that for each $\hat{\omega}$ the constants of each $S_i$ are distributed among the truth configurations in $\Theta$. Note that $|\Theta| = 2^{|Pred|}$. A distribution of constants of $S_i$ among $\Theta$ can be combined with any distribution of constants of $S_j$ for every $i \neq j$, yielding a single reference world $\hat{\omega}$. In order to count the number of reference worlds we need to multiply the number of combinations one can distribute the constants of $S_i$ onto the truth configurations in $\Theta$ with the number of combinations for every other $S_j$ $(i \neq j)$. Then we get

$$|\hat{\Omega}| = \prod_{i=1}^{n} |\{(l_1, \ldots, l_{2^{|Pred|}}) \in \mathbb{N}_0^{2^{|Pred|}} \mid l_1 + \ldots + l_{2^{|Pred|}} = |S_i|\}| \quad . \tag{8}$$

Each factor in the product of the above equation represents the number of combinations the constants of a single $\mathcal{R}$-equivalence class can be distributed among the possible truth configurations in $\Theta$. The condition $l_1 + \ldots + l_{2^{|Pred|}} = |S_i|$ ensures that each constant is exactly assigned one truth configuration in every combination. Still, Equation (8) gives no direct hint on the space needed to represent $\hat{\Omega}$ in terms of $|U_\Sigma|$ and $|Pred|$. But it is possible to rewrite (8) as follows.

**Definition 12.** *The* cardinality generator $g_c$ *is the function* $g_c : \mathbb{N}_0^2 \to \mathbb{N}_0$ *defined via*

$$g_c(n_1, n_2) =_{def} \begin{cases} \sum_{i=0}^{n_2} g_c(n_1 - 1, n_2 - i) & \text{if } n_2 > 0 \text{ and } n_1 > 0 \\ 1 & \text{if } n_2 = 0 \\ 0 & \text{otherwise} \end{cases} \quad .$$

The intuition behind using $g_c$ to enumerate the number of reference worlds is as follows. The first argument of $g_c$ is meant to represent the number of truth configurations and the second the number of constants in an $\mathcal{R}$-equivalence class. By defining $g_c(n_1, n_2) = g_c(n_1 - 1, 0) + \ldots + g_c(n_1 - 1, n_2)$ we say that the number of combinations to distribute $n_2$ constants on $n_1$ truth configuration is equal to the number of combinations to distribute zero constants on $n_1 - 1$ truth configurations plus the number of combinations to distribute one constant on $n_1 - 1$ truth configurations, and so on. The first case describes a setting where we assign all

$n_2$ constants to the $n_1$th truth configuration and as there are no remaining constants left this amounts to the number of $g_c(n_1 - 1, 0)$ remaining combinations. The second case describes a setting where we assign $n_2 - 1$ constants to the $n_1$th truth configuration and the remaining single constant to the remaining $n_1 - 1$ truth configurations. The final case describes the setting of assigning no constant the $n_1$th truth configuration and the remaining $n_2$ constants to the remaining $n_1 - 1$ truth configurations. Consider $g_c(1, 3)$ as the number of combinations to distribute three constants on one truth configuration. Applying the first case of the definition of $g_c$ yields $g_c(1, 3) = g_c(0, 0) + g_c(0, 1) + g_c(0, 2) + g_c(0, 3)$ and therefore the number of combinations to distribute three constants on one truth configuration is to assign all three constants to the one truth configuration, or to assign zero, one, or two to it. Obviously, the latter cases are not valid and the only valid assignment is that three constants are assigned to the one truth configuration. Due to the third case in the definition of $g_c$ the terms $g_c(0, 1)$, $g_c(0, 2)$, and $g_c(0, 3)$ are set to zero.

**Proposition 8.** *It holds that*

$$|\hat{\Omega}| = \prod_{i=1}^{n} g_c(2^{|Pred|}, |S_i|) \quad . \tag{9}$$

Still, Equation (9) does not allow to get an idea of the size of $|\hat{\Omega}|$. However, the function $g_c$ can be bounded from above as follows.

**Lemma 1.** *It holds that* $g_c(n_1, n_2) \leq (n_2 + 1)^{n_1}$ *for every* $n_1, n_2 \in \mathbb{N}_0$.

**Theorem 2.** *It holds that* $|\hat{\Omega}| \leq (|\mathsf{Const}(\mathcal{R})| + 1)(|U_\Sigma| + 1)^{2^{|Pred|}}$.

The obvious observation to be made when comparing $|\Omega(\Sigma)|$ to the upper bound of $|\hat{\Omega}|$ is that the latter is not exponential in the number of constants $|U_\Sigma|$. But note that the complexity increases with respect to $|Pred|$. While $|\Omega(\Sigma)|$ is exponential in $|Pred|$, the above bound for $|\hat{\Omega}|$ is exponential in $2^{|Pred|}$. However, we believe that this is due to a very coarse estimation in Lemma 1. Experiments suggest that $g_c$ can be much better estimated.

*Conjecture 1.* It holds that $g_c(n_1, n_2) \leq (n_2 + 1)^{2^{\mathsf{Id}\, n_1}}$ (with $\mathsf{Id}\, 0 = 0$).

The above conjecture would result in an upper bound of $(|\mathsf{Const}(\mathcal{R})| + 1)(|U_\Sigma| + 1)^{2^{|Pred|}}$ which is far more beneficial than the result of Theorem 2. However, until now no formal proof for the above conjecture has been found.

Table 1 shows some exemplary cardinalities of $\Omega(\Sigma)$ and $\hat{\Omega}$ for different values of $|U_\Sigma|$ and $|Pred|$. The knowledge base $\mathcal{R}$ used to determine the $\mathcal{R}$-equivalences classes in $\hat{\Omega}$ mentions a single constant yielding $\mathfrak{S}(\mathcal{R}) = \{\{\mathsf{c}\}, U_\Sigma \setminus \{\mathsf{c}\}\}$ for $\mathsf{Const}(\mathcal{R}) = \{\mathsf{c}\}$. Table 1 shows that especially for this kind of scenarios employing $\hat{\Omega}$ rather than $\Omega(\Sigma, D)$ is computationally beneficial. The numbers in Table 1 also justify the belief in Conjecture 1.

## 6 Generalizing Lifted Inference

In contrast to the case without non-unary predicates there is no simple and compact representation of $\Omega(\Sigma)/_{\equiv_\mathcal{R}}$ if $\Sigma$ contains at least one non-unary predicate

| $\lvert Pred\rvert$ | $\lvert U_\Sigma\rvert$ | $\lvert\Omega(\Sigma)\rvert$ | $\lvert\hat{\Omega}\rvert$ | $\lvert Pred\rvert$ | $\lvert U_\Sigma\rvert$ | $\lvert\Omega(\Sigma)\rvert$ | $\lvert\hat{\Omega}\rvert$ |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 4 | 4 | 2 | 2 | 16 | 16 |
| 1 | 8 | 256 | 16 | 2 | 4 | 256 | 64 |
| 1 | 32 | 4294967296 | 64 | 2 | 8 | 65536 | 256 |

**Table 1.** Comparison of $|\Omega(\Sigma)|$ and $|\hat{\Omega}|$ with respect to a knowledge base $\mathcal{R}$ with $|\mathsf{Const}(\mathcal{R})| = 1$

and, in particular, no compact way to enumerate the elements of $\Omega(\Sigma)/_{\equiv_\mathcal{R}}$. Consider a predicate $p/2$ and $\mathcal{R}$-equivalence classes $S_1$ and $S_2$. Then there are six different instantiations of $p$ that have to be considered as essentially different with respect to $\mathcal{R}$-equivalence. For constants $\mathsf{c}_1 \in S_1$ and $\mathsf{c}_2 \in S_2$ we have the variants $p(\mathsf{c}_1, \mathsf{c}_2)$ and $p(\mathsf{c}_2, \mathsf{c}_1)$; for $\mathsf{c}_1 \in S_i$ we have $p(\mathsf{c}_1, \mathsf{c}_1)$ for $i = 1, 2$; for $\mathsf{c}_1, \mathsf{c}_2 \in S_i$ with $\mathsf{c}_1 \neq \mathsf{c}_2$ we have $p(\mathsf{c}_1, \mathsf{c}_2)$ for $i = 1, 2$. An extended notion of truth configuration must adhere to this combinatorial observation and also take the relations into account that arise by transitivity. In the unary case, we used truth configurations to be able to enumerate the elements of $\Omega(\Sigma)/_{\equiv_\mathcal{R}}$ in an effective way without considering $\Omega(\Sigma)$ itself. In the non-unary case there seems to be no simple way to extend the concept of truth configuration. This observation has also been made by Grove et. al. in [5] when they attempted to generalize the notion of entropy of an interpretation to non-unary languages, see [5] on page 67 for a discussion.

However, the approach of lifted inference developed in this chapter can be applied for non-unary languages by determining first $\Omega(\Sigma)$ and afterwards (by pair-wise comparisons) merge $\mathcal{R}$-equivalent interpretations to reference worlds (yielding the quotient set $\Omega(\Sigma)/_{\equiv_\mathcal{R}}$). Note that we lose the computational advantage of avoiding to consider the full set $\Omega(\Sigma)$ in this approach. It is also questionable whether using $\Omega(\Sigma)/_{\equiv_\mathcal{R}}$ instead of $\Omega(\Sigma)$ for inference is beneficial. Table 2 shows the cardinalities of both $\Omega(\Sigma)$ and $\Omega(\Sigma)/_{\equiv_\mathcal{R}}$, depending on the size of $U_\Sigma$ and with respect to a signature containing a single binary predicate and a knowledge base $\mathcal{R}$ with $\mathsf{Const}(\mathcal{R}) = \emptyset$. As $\mathcal{R}$ mentions no constants there is only one single $\mathcal{R}$-equivalence class which makes this scenario the simplest imaginable. Nonetheless, the cardinality of $\Omega(\Sigma)/_{\equiv_\mathcal{R}}$—although being significantly smaller than the cardinality of $\Omega(\Sigma)$—still seems to grow exponentially in the number of constants considered. Until now, no formal proofs for lower or upper bounds on the growing behavior of $|\Omega(\Sigma)/_{\equiv_\mathcal{R}}|$ have been found. However, Table 2 gives reason to believe that there is no polynomial upper bound for $|\Omega(\Sigma)/_{\equiv_\mathcal{R}}|$ in $|U_\Sigma|$. As a consequence, lifted inference in RPCL can be doubted to be beneficial at all for non-unary languages.

## 7 Related Work

The notion of *lifted inference* used in this paper has been adopted from the works [11,13,8] which also use this notion to describe effective reasoning procedures for

| $\lvert U_\Sigma\rvert$ | $\lvert\Omega(\Sigma)\rvert$ | $\lvert\Omega(\Sigma)/_{\equiv_\mathcal{R}}\rvert$ | $\lvert U_\Sigma\rvert$ | $\lvert\Omega(\Sigma)\rvert$ | $\lvert\Omega(\Sigma)/_{\equiv_\mathcal{R}}\rvert$ |
|---|---|---|---|---|---|
| 1 | 2 | 2 | 3 | 512 | 244 |
| 2 | 16 | 10 | 4 | 65536 | 12235 |

**Table 2.** Comparison of $\lvert\Omega(\Sigma)\rvert$ and $\lvert\Omega(\Sigma)/_{\equiv_\mathcal{R}}\rvert$ with respect to a signature that contains a single binary predicate and a knowledge base $\mathcal{R}$ with $\mathsf{Const}(\mathcal{R}) = \emptyset$

relational probabilistic knowledge, see also [1,4] for some recent work.. Although the knowledge representation formalisms of those approaches differ to our approach, the motivation and ideas of those approaches are similar to ours. The work [13]—which extends work begun in [11]—develops an algorithm for lifted probabilistic inference in parametrized belief networks. The basic idea of [11,13] is the observation that in order to determine the probability of some query the information used to infer the probability can be partitioned with respect to the information we have for specific individuals. This approach uses the technique of *variable elimination* to simplify computation of probabilities with respect to equivalencies of undistinguishable constants. We do not give a formal description of the algorithms developed in [11,13] but rather give an idea of the approach by means of an example. Consider the clause $c =_{def} (p(\mathsf{X}) \mid q(\mathsf{X},\mathsf{Y}), r(\mathsf{Y}))$ and a function $\mathsf{cpd}_c$ (conditional probability distribution) which maps each possible truth configuration to a probability, e. g. $\mathsf{cpd}_c(\mathsf{true},\mathsf{true},\mathsf{false}) = 0.7$ states that the probability of observing $p(\mathsf{c}_1)$ given that $q(\mathsf{c}_1,\mathsf{c}_2)$ is true and $r(\mathsf{c}_2)$ is false is 0.7 (for all constant symbols $\mathsf{c}_1,\mathsf{c}_2$). Note that $c$ can be instantiated using different assignments for $\mathsf{Y}$ but with the same $\mathsf{X}$. In this case, one can employ a combining rule such as *noisy-or* [10] to aggregate probabilities, i. e., the *noisy-or* of two probabilities $p_1$ and $p_2$ is defined as $1-(1-p_1)(1-p_2)$. Let now $E_{n,m} =_{def} \{q(\mathsf{c},\mathsf{d}_1),\ldots,q(\mathsf{c},\mathsf{d}_{n+m})\} \cup \{r(\mathsf{d}_1),\ldots,r(\mathsf{d}_n),\neg r(\mathsf{d}_{n+1}),\ldots,\neg r(\mathsf{d}_{n+m})\}$ be some observed evidence with $n,m \in \mathbb{N}$ and consider determining the probability $P(p(\mathsf{c}) \mid E)$. In e. g. ordinary BLPs [3, Ch. 10] one has to instantiate a ground Bayesian network for the node $p(\mathsf{c})$ with parents $q(\mathsf{d}_1),\ldots,q(\mathsf{d}_{n+m}), r(\mathsf{c},\mathsf{d}_1),\ldots,r(\mathsf{c},\mathsf{d}_{n+m})$, and combine the probabilities using *noisy-or*. This amounts to

$$\begin{aligned} P(p(\mathsf{c}) \mid E) = 1 - (1 - P(p(\mathsf{c}) \mid q(\mathsf{c},\mathsf{d}_1), r(\mathsf{d}_1))) \cdot \ldots \cdot \\ (1 - P(p(\mathsf{c}) \mid q(\mathsf{c},\mathsf{d}_{n+m}), r(\mathsf{d}_{n+m}))) \quad . \end{aligned}$$

Note that we have the same information for the constant symbols $\mathsf{d}_1,\ldots,\mathsf{d}_n$ and $\mathsf{d}_{n+1},\ldots,\mathsf{d}_m$, respectively. It follows that

$$\begin{aligned} P(p(\mathsf{c}) \mid q(\mathsf{c},\mathsf{d}_1), r(\mathsf{d}_1)) &= \ldots = P(p(\mathsf{c}) \mid q(\mathsf{c},\mathsf{d}_n), r(\mathsf{d}_n)) \\ &= \mathsf{cpd}_c(\mathsf{true},\mathsf{true},\mathsf{true}) \\ P(p(\mathsf{c}) \mid q(\mathsf{c},\mathsf{d}_{n+1}), r(\mathsf{d}_{n+1})) &= \ldots = P(p(\mathsf{c}) \mid q(\mathsf{c},\mathsf{d}_{n+m}), r(\mathsf{d}_{n+m})) \\ &= \mathsf{cpd}_c(\mathsf{true},\mathsf{true},\mathsf{false}) \end{aligned}$$

and therefore

$$P(p(\mathsf{c}) \mid E) = 1 - (1 - \mathsf{cpd}_c(\mathsf{true},\mathsf{true},\mathsf{true}))^n (1 - \mathsf{cpd}_c(\mathsf{true},\mathsf{true},\mathsf{false}))^m \ .$$

As one can see, we can avoid grounding the full BLP by just considering *prototypical* groundings for $c$. In [11,13] this idea is elaborated and a series of algorithms is developed that apply this approach to general parametrized belief networks (or BLPs). Obviously, the ideas of [11,13] are very similar to ours and differences lie mainly on the framework used for knowledge representation and the technical implementation. The work [11] uses parametrized belief networks and inference bases on Bayesian networks and [13] uses a framework similar to MLNs [3, Ch. 12]. However, note that both formalisms are first-order extensions of probabilistic networks but we use RPCL and inference based on the principle of maximum entropy. Furthermore, we developed an explicit computational model for representing prototypical uniform probability functions and showed that the use of this model is beneficial in terms of computational complexity. In [11] no hints on the computational advantages of applying first-order variable elimination are given but [13] gives an experimental evaluation that resembles our observations from Conjecture 1.

## 8 Summary and Conclusion

We developed a computational account for effective probabilistic inference with relational probabilistic conditionals. In particular, we introduced the notions of reference worlds and condensed probability functions which allow for a compact representation of probability functions that arise from the application of inference operators satisfying *prototypical indifference*. Condensed probability functions are defined on the set of reference worlds and exhibit the same reasoning behavior as the original probability functions, given that those are indifferent with respect to constants from the same $\mathcal{R}$-equivalence class. Furthermore, we showed that the inference operators from [6] can be modified in order to compute the condensed maximum entropy function in a single step without considering the Herbrand interpretations at all. We analyzed the computational benefits of our approach and concluded that we avoid the exponential blow-up in the number of constants that have to be considered. Our approach is—using the given formalization—only applicable for unary languages and we briefly discussed the issues that arise when considering non-unary languages.

The approach developed in this paper gives some first directions for efficient implementation of reasoning based on the principle of maximum entropy. However, the work reported is only a first step towards this goal and suffers from two major discrepancies. Firstly, we restricted lifted inference to the case of unary languages which, in practice, is a demand that cannot be easily fulfilled. One of the main advantages of first-order extensions of probabilistic reasoning is the capability to reason over relations. However, note that even by restricting attention to unary languages we do not get the equivalence to propositional probabilistic models due to our semantical notions. For example, the knowledge base $\mathcal{R} =_{def} \{(\mathit{flies}(\mathsf{X}))[0.9], (\mathit{flies}(\mathsf{tweety}))[0.3]\}$ cannot be represented using a propositional probabilistic model that exhibits the same inference behavior. Secondly, in order to determine the (condensed) maximum entropy model of a knowledge base $\mathcal{R}$ we have to solve a complex optimization problem. However,

there are approaches to avoid solving problems like (3) for the propositional case. For example, in [7] an approximate algorithm for computing the maximum entropy model for propositional probabilistic conditional logic is developed. The algorithm of [7] benefits from several characteristic properties of the maximum entropy model in the propositional case and it is to investigate if these properties (or similar ones) can be found for our semantical approaches which may lead to the development of algorithms for approximate inference in relational probabilistic conditional logic.

**Acknowledgements.** The research reported here was partially supported by the Deutsche Forschungsgemeinschaft (grant KE 1413/2-1).

## References

1. van den Broeck, G., Taghipour, N., Meert, W., Davis, J., Raedt, L.D.: Lifted Probabilistic Inference by First-Order Knowledge Compilation. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11). pp. 2178–2185 (2011)
2. Delgrande, J.P.: On first-order conditional logics. Artificial Intelligence 105(1-2), 105–137 (1998)
3. Getoor, L., Taskar, B. (eds.): Introduction to Statistical Relational Learning. MIT Press (2007)
4. Gogate, V., Domingos, P.: Probabilistic theorem proving. In: Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI'11) (2011)
5. Grove, A.J., Halpern, J.Y., Koller, D.: Random worlds and maximum entropy. Journal of Artificial Intelligence Research (JAIR) 2, 33–88 (1994)
6. Kern-Isberner, G., Thimm, M.: Novel semantical approaches to relational probabilistic conditionals. In: Proceedings of KR'10 (2010)
7. Meyer, C.H.: Korrektes Schliessen bei unvollständiger Information. Ph.D. thesis, FernUniversität in Hagen, Germany (1997)
8. Milch, B., Zettlemoyer, L.S., Kersting, K., Haimes, M., Kaelbling, L.P.: Lifted Probabilistic Inference with Counting Formulas. In: Procedings of AAAI'08 (2008)
9. Paris, J.B.: The Uncertain Reasoner's Companion – A Mathematical Perspective. Cambridge University Press (1994)
10. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1998)
11. Poole, D.: First-Order Probabilistic Inference. In: Proceedings of IJCAI'03. pp. 985–991. Morgan Kaufmann (2003)
12. Rödder, W.: Conditional logic and the principle of entropy. Artificial Intelligence 117, 83–106 (2000)
13. de Salvo Braz, R., Amir, E., Roth, D.: Lifted First-Order Probabilistic Inference. In: Proceedings of IJCAI'05. pp. 1319–1325 (2005)
14. Thimm, M.: Probabilistic Reasoning with Incomplete and Inconsistent Beliefs. Ph.D. thesis, Technische Universität Dortmund, Germany (2011), submitted
15. Thimm, M., Kern-Isberner, G., Fisseler, J.: Relational probabilistic conditional reasoning at maximum entropy. In: Proceedings of ECSQARU'11 (2011)
16. Wellman, M.P., Breese, J.S., Goldman, R.P.: From Knowledge Bases to Decision Models. The Knowledge Engineering Review 7(1), 35–53 (1992)

# Knowledge Engineering with Markov Logic Networks: A Review

Dominik Jain

Intelligent Autonomous Systems Group
Technische Universität München
jain@cs.tum.edu

**Abstract.** Within the realm of statistical relational knowledge representation formalisms, Markov logic is perhaps one of the most flexible and general languages, for it generalises both first-order logic (for finite domains) and probabilistic graphical models. Knowledge engineering with Markov logic is, however, not a straightforward task. In particular, modelling approaches that are too firmly rooted in the principles of logic often tend to produce unexpected results in practice. In this paper, we collect a number of issues that are relevant to knowledge engineering practice: We describe the fundamental semantics of Markov logic networks and explain how simple probabilistic properties can be represented. Furthermore, we discuss fallacious modelling assumptions and summarise conditions under which generalisation across domains may fail. As a collection of fundamental insights, the paper is primarily directed at knowledge engineers who are new to Markov logic.

## 1 Introduction

In artificial intelligence (AI), the unification of statistical and relational knowledge within a single representation formalisms is an important line of research, for it addresses two of the most pressing needs of AI systems: the ability to deal with the uncertainty inherent in real-world domains and the complex interactions between entities that are relevant to an agent, which can adequately be described using relations. In recent years, we have seen tremendous advances in the field that has emerged as statistical relational learning (SRL) and reasoning [2]. One of the most fundamental principles of knowledge representation formalisms developed in SRL is generalisation across domains (shallow transfer), i.e. models are to represent not a concrete probability distribution but rather a set of general principles that can be applied to an arbitrary set of entities and the relations between them in order to obtain a concrete probabilistic model.

A representation formalism that has garnered much attention – owing to its generality and conceptual simplicity – is Markov logic [7], which generalises both first-order logic and probabilistic graphical models (Markov networks) by attaching weights to formulas in first-order logic. Collectively, the weighted formulas define a template for the construction of a graphical model that specifies a distribution over possible worlds. In essence, the probability of a possible world

is proportional to the exponentiated sum of weights of formulas that are satisfied in that world.

As an example, consider the following Markov logic network (MLN),

$$
\begin{array}{ll}
w_1 = 0.000 & \forall p.\ \mathit{female}(p) \\
w_2 = -1.735 & \forall p.\ \mathit{vegetarian}(p) \\
w_3 = 1.400 & \forall p.\ \mathit{female}(p) \Rightarrow \mathit{vegetarian}(p) \\
w_4 = 1.300 & \forall p_1, p_2.\ \mathit{friends}(p_1, p_2) \wedge \mathit{vegetarian}(p_1) \Rightarrow \mathit{vegetarian}(p_2) \\
w_5 = 2.300 & \forall p, d.\ \mathit{orders}(p, d) \wedge \neg \mathit{vegetarian}(p) \Rightarrow \neg \mathit{vegetarianDish}(d) \\
w_6 = (\text{hard}) & \forall p, d.\ \mathit{orders}(p, d) \wedge \mathit{vegetarian}(p) \Rightarrow \mathit{vegetarianDish}(d)
\end{array}
$$

which represents a scenario where people go to a restaurant to order a dish. In addition to the weighted formulas themselves, MLNs typically contain declarations that define the types of entities to which predicates can be applied. In our scenario, the predicates are applicable to *person* and *dish* entities. Furthermore, predicates/relations can be declared as being functional, as this is a particularly common requirement in relational domains. In our example, *orders* is declared as functional, such that every person is required to order exactly one dish.

The weights in the MLN determine the degree to which the formulas they are attached . By attaching a zero weight to the first formula, we essentially make no statement about whether being female or not being female is more probable. Most people, however, are not vegetarians; hence the negative weight $w_2$. The positive weight $w_3$ is intended to express that females are more likely to be vegetarians. The fourth formula considers a social network structure defined by the *friends* relation and states that friends of vegetarians are more likely to themselves be vegetarians. The last two formulas express consumption preferences: Non-vegetarians are inclined to order non-vegetarian dishes; vegetarians always order vegetarian dishes. The latter constraint is hard, meaning that any world that does not satisfy the formula is to have zero probability.

The above MLN seems to be an intuitive representation of this very simple scenario. However, it displays some perhaps unexpected behaviour. First, we observe that, for any given domain of people and dishes, the probability of being female is not 0.5, yet *female* appears only in the first formula, which expresses no inclination in either direction, and in the antecedent of an implication (formula 3). Further, the probability is not only far from 0.5, it also varies depending on the number of people and dishes in the domain. For instance, if there are two dishes that can be ordered, then the probability of being female for cases where the number of persons is one, two and three, we obtain approximately[1] 0.223, 0.211 and 0.204 respectively. For *vegetarian* and *vegetarianDish*, we observe particularly wide variations: The probabilities for a dish being vegetarian are 0.182, 0.120 and 0.07 and the probabilities for a person being a vegetarian are 0.084, 0.044 and 0.020 respectively. While at least some of these observations are certainly obvious if one ponders the underlying mathematical definitions, they suggest that a careful consideration of MLN semantics is the very foundation of any knowledge engineering endeavour.

[1] All the inference results reported in this paper were computed with exact methods. Where an approximation is mentioned, it refers to the result being rounded only.

The mathematical description of the semantics of Markov logic networks is deceivingly simple, yet the task of knowledge engineering in Markov logic networks is, as indicated by our example, not as straightforward as it may at first appear. The question of how one could manually define a Markov logic network remains largely unaddressed in the literature. Indeed, learning is often the only way to sensibly derive suitable parameters for an MLN. However, we believe that a thorough understanding of how we *could*, in principle, represent particular probabilistic properties in an MLN is imperative to selecting the formulas that are in fact required for a model to at all be able to reflect the desired properties and obtain a model that is, ideally, an adequate reflection of the real world. At present, there is no collection of guidelines that adequately addresses the problem of defining weighted formulas for cases where a manual specification might still be feasible. With this paper, we make the following contributions:

- We provide an in-depth discussion of how simple probabilistic properties can be represented in an MLN.
- We point out fallacious assumptions that result from a naive interpretation of the nature of the transition from logical to probabilistic knowledge. In particular, we discuss the fallacious use of probabilistic implications and describe how the intended pieces of knowledge can be represented by other means.
- We summarise practical issues pertaining to the way in which models in Markov logic – be they learnt from data or manually defined – generalise across domains.

## 2 Markov Logic Networks

Formally, an MLN $L$ is given by a set of pairs $\langle F_i, w_i \rangle$ [7], where $F_i$ is a formula in first-order logic and $w_i$ is a real-valued weight. For each finite domain of discourse $D$ (set of constants), an MLN $L$ defines a *ground Markov network* $M_{L,D} = \langle X, G \rangle$ as follows:

1. $X$ is a set of boolean variables. For each possible grounding of each predicate appearing in $L$, we add to $X$ a boolean variable (ground atom). We denote by $\mathcal{X} := \mathbb{B}^{|X|}$ the set of possible worlds, i.e. the set of possible assignments of truth values to the variables in $X$ (see Fig. 1).
2. $G$ is a set of weighted ground formulas, i.e. a set of pairs $\langle \hat{F}_j, \hat{w}_j \rangle$, where $\hat{F}_j$ is a ground formula and $\hat{w}_j$ is a real-valued weight. For each possible grounding $\hat{F}_j$ of each formula $F_i$ in $L$, we add to $G$ the pair $\langle \hat{F}_j, \hat{w}_j = w_i \rangle$. With each such pair, we associate a *feature* $\hat{f}_j : \mathcal{X} \to \{0, 1\}$, whose value for $x \in \mathcal{X}$ is 1 if $\hat{F}_j$ is satisfied in $x$ and 0 otherwise, and whose weight is $\hat{w}_j$.

In practice, we typically use a typed logic and augment the above definition with declarations that define the types of entities a predicate applies to. Furthermore, a particularly common real-world restriction is to require relations to be functional and MLN implementations support the corresponding declarations as an extension.[2]

[2] For instance, the declaration *orders*(*person*, *dish*!) states that the predicate *orders* is applicable to *person* and *dish* entities and that for each person, there must be exactly

The ground Markov network $M_{L,D}$ specifies a probability distribution over the set of possible worlds $\mathcal{X}$ as follows,

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_{i=1}^{|L|} w_i n_i(x)\right) = \frac{1}{Z} \exp\left(\sum_{j=1}^{|G|} \hat{w}_j \hat{f}_j(x)\right) \tag{1}$$

where $Z = \sum_{x' \in \mathcal{X}} \exp(\sum_i w_i n_i(x')) = \sum_{x' \in \mathcal{X}} \exp(\sum_j \hat{w}_j \hat{f}_j(x'))$ is a normalization constant and $n_i(x)$ denotes the number of true groundings of $F_i$ in $x$.

The probability of a possible world $x \in \mathcal{X}$ is thus proportional to the exponentiated sum of weights of formulas that are satisfied in $x$, i.e.

$$P(x) \propto \exp\left(\sum_j \hat{w}_j \hat{f}_j(x)\right) =: \omega(x). \tag{2}$$

With $s(F) := \sum_{x \in \mathcal{X},\, x \models F} \omega(x)$, we can calculate the probability of any ground formula $F_1$ given any other ground formula $F_2$ as

$$P(F_1 \mid F_2) = \frac{P(F_1, F_2)}{P(F_2)} = \frac{s(F_1 \wedge F_2)}{s(F_2)}. \tag{3}$$

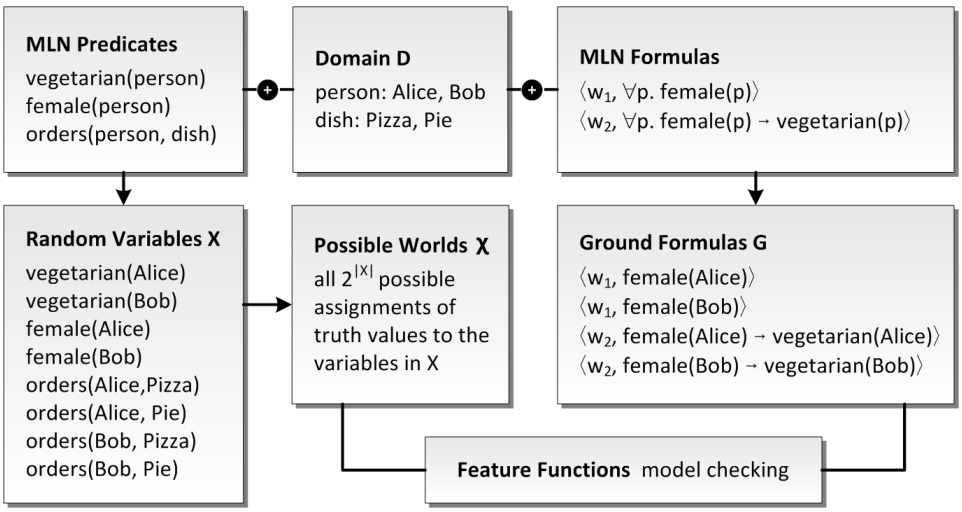


**Fig. 1.** Illustration of the grounding process

## 3 Fundamental Semantics

We can differentiate two ways in which MLN semantics can be viewed:

(V1) the top-down view, in which we seek to realise a probabilistic logic by adding weights to formulas; Markov networks serve only as a formalism that provides the particular probabilistic semantics;

one dish for which the predicate holds. Any possible world violating this functional property of the relation has zero probably.

(V2) the bottom-up view, in which MLNs are an extension of the Markov network formalism and weighted formulas are a particular way of compactly representing generalised features of Markov networks; Any ground formula defines a clique in the ground Markov network and the formula's set of models (satisfying assignments) determines the subset of the clique's configurations that is affected.

From the perspective of artificial intelligence, the top-down view is perhaps more attractive, because the generalisation of logic to probabilistic settings was a long-standing goal of the community. Probability theory is widely accepted as the preferred way of dealing with the inherent uncertainty that permeates real-world domains, and logics are well-understood formalisms for the representation of complex, structured knowledge. To seek a sound unification is, indeed, the merest step of logic.

However, it is the bottom-up view that is imperative to a deep understanding of MLNs. As will be shown in this report, a thorough understanding of Markov networks and, in particular, the way in which weighted formulas translate to features in Markov networks is the very foundation for knowledge engineering in MLNs. Unfortunately, a modelling approach that is too firmly rooted in the principles of logic will often translate sub-optimally to probabilistic settings.

### 3.1 Quantifier Semantics

A logical knowledge base (KB) is the conjunction of all the formulas that are known to hold. In an MLN, we can choose to partially soften the KB and attach weights to arbitrary parts of the full conjunction, weights indicating the degree to which the individual formulas are required to hold. The granularity at which we apply weighting can be selected as necessary.

**The Universal Quantifier for Parameter Sharing.** In this context, the semantics of the universal quantifier are particularly relevant. In the logical case, a formula such as $\forall e.\ \mathit{apple}(e) \Rightarrow \mathit{fruit}(e)$ translates, for a finite domain of discourse $D$, to the conjunction $\bigwedge_{E \in D} \mathit{apple}(E) \Rightarrow \mathit{fruit}(E)$. In MLNs, however, the weighted formula $\langle w, \forall e.\ \mathit{apple}(e) \Rightarrow \mathit{fruit}(e) \rangle$ does not result in the weight $w$ being attached to the conjunction but rather in the weight being attached to each conjunct separately, i.e. its semantics can be understood as $\forall E \in D.\ (\langle w, \mathit{apple}(E) \Rightarrow \mathit{fruit}(E) \rangle \in G)$. Thus, the universal quantifier actually serves to realise parameter sharing. Parameter sharing is a fundamental principle in relational models, which serves to achieve the generalisation across arbitrary domains, for it enables us to apply the same probabilistic patterns to any given set of entities (shallow transfer).

In most implementations of MLNs, a universal quantifier at the outermost level can be omitted – and, given its semantics, its omission may even serve to improve clarity. We consequently chose to omit such universal quantifiers in the remainder of this document.

**Existential Quantification.** Notably, the existential quantifier does not behave analogously; it expands to a single disjunction with weight $w$ (as does a universal

quantifier at an inner level). Hence disjunctions are always evaluated strictly logically. Since KBs are "naturally" dissected into conjunctions, we do not consider this to be a practically relevant restriction, however. Nevertheless, the formalism of recursive random fields [6] was proposed to also allow disjunctions to be evaluated in a "soft" manner.

### 3.2 Formula Weights

**Weights as logarithmized factors.** To determine the precise effect of a weight in an MLN, we need only to consider the impact that weights have on the joint distribution over $\mathcal{X}$ (for an arbitrary but fixed grounding). Every non-zero weight $w_i$ of every formula $F_i$ affects the sum $\sum_i w_i n_i(x)$ for some subset of $\mathcal{X}$ and thus has bearing on the distribution. Because $P(X = x)$ is proportional to the *exponentiated* sum of weights that are true in world $x$, weights are best viewed as logarithmized factors. In the absence of formulas, every possible world's value $\omega(x)$ is $\exp(0) = 1$ and therefore all worlds are equally probable. If we now add the pair $\langle F_i, w_i = \log(\lambda_i)\rangle$ to $L$ and some possible world $x$ satisfies a ground instance $\hat{F}_j$ of $F_i$, then $\omega(x)$ is simply multiplied by $\exp(w_i) =: \lambda_i = \hat{\lambda}_j$, either increasing the probability of $x$ (for $\lambda_i > 1 \Leftrightarrow w_i > 0$) or decreasing it (for $\lambda_i \in ]0;1[ \Leftrightarrow w_i < 0$). Correspondingly,

$$P(X = x) \propto \prod_j \exp(\hat{f}_j(x)\hat{w}_j) = \prod_{j,\, x \models \hat{F}_j} \hat{\lambda}_j. \tag{4}$$

When interpreting the semantics of a particular weight $w_i$ of a particular formula $F_i$, it is imperative to be aware of the factor it implies and the subset of $\mathcal{X}$ it will be applied to (for any imaginable instantiations of $L$). The impact of a particular formula is perhaps most clearly evident if the formula is translated to disjunctive normal form (DNF), because the conjunctions therein immediately reflect partial assignments in possible worlds. For instance, all of the following are semantically equivalent,

$$\begin{array}{ll} w & \mathit{bird}(e) \Rightarrow \mathit{flies}(e) \\ w & \neg \mathit{bird}(e) \vee \mathit{flies}(e) \\ w & (\mathit{bird}(e) \wedge \mathit{flies}(e)) \vee (\neg \mathit{bird}(e) \wedge \mathit{flies}(e)) \vee (\neg \mathit{bird}(e) \wedge \neg \mathit{flies}(e)) \\ -w & \mathit{bird}(e) \wedge \neg \mathit{flies}(e) \end{array}$$

yet the latter two forms perhaps more clearly indicate the variable assignments that are affected. In the fourth case, we negated the formula along with its weight to obtain a more easily interpretable conjunction.[3]

[3] An MLN retains its semantics if any of its formulas are negated along with their weights, i.e. the probability distributions implied by an MLN $L$ and an MLN $L'$ derived from $L$ by changing a pair $\langle F_k, w_k\rangle \in L$ to $\langle \neg F_k, -w_k\rangle \in L'$ are the same for all ground models. If $L$ and $L'$ are instantiated for a set of constants $C$, for which $F_k$ has $N_k$ groundings, we have

$$P_{M_{L',C}}(x) = \frac{\exp\left(\sum_{i,i\neq k} w_i \cdot n_i(x) - w_k \cdot (N_k - n_k(x))\right)}{\sum_{x' \in \mathcal{X}} \exp\left(\sum_{i,i\neq k} w_i \cdot n_i(x') - w_k \cdot (N_k - n_k(x'))\right)}$$

$$= \frac{\exp\left(\sum_i w_i \cdot n_i(x) - w_k \cdot N_k\right)}{\sum_{x' \in \mathcal{X}} \exp\left(\sum_i w_i \cdot n_i(x') - w_k \cdot N_k\right)} = \frac{\exp\left(\sum_i w_i \cdot n_i(x)\right)}{\sum_{x' \in \mathcal{X}} \exp\left(\sum_i w_i \cdot n_i(x')\right)} \cdot \frac{\cancel{\exp(-w_k \cdot N_k)}}{\cancel{\exp(-w_k \cdot N_k)}} = P_{M_{L,C}}(x)$$

While the local view of the effect of a formula weight on the values $\omega(x)$ associated with possible worlds sufficiently describes the semantics of MLNs, it is, for the most part, not very helpful from a knowledge engineering perspective. In particular, it does not immediately answer the question of how we could define weights that will result in the model reflecting a particular probabilistic property.

**Weights as log odds between world probabilities.** To determine the effect of a particular weight in terms of probability, let us consider a single (ground) formula $\hat{F}_j$ with an associated weight $w_i = \hat{w}_j$ and let us assume that the MLN contains no further formulas. As mentioned in [7], we can, in such a case, interpret $w_j$ as the log odds between the probability of a world where $\hat{F}_j$ is true and the probability of a world where $\hat{F}_j$ is false. Thus, if $x_1 \models \hat{F}_j$ and $x_2 \not\models \hat{F}_j$ ($x_1, x_2 \in \mathcal{X}$), then $P(x_1) : P(x_2) = \hat{\lambda}_j : 1$, since $\hat{F}_j$ being true will result in a factor of $\hat{\lambda}_j$ being applied to worlds satisfying $\hat{F}_j$ while worlds not satisfying $\hat{F}_j$ will keep the default factor of 1. Setting $\hat{w}_j = \log(P(x_1)/P(x_2))$ will thus establish the proper ratio between worlds satisfying $\hat{F}_j$ and worlds not satisfying $\hat{F}_j$.

**Weights that determine formula probabilities.** Regarding the probabilistic properties that are to be represented in our model, the local view of ratios between probabilities of individual worlds is clearly inadequate. We need to consider the global impact on all possible worlds in order to asses the properties of the distribution $P(x)$ that are induced by a particular choice of weights, which, in particular, necessitates taking model counts into consideration.

The perhaps most basic modelling task is to define a single formula $F_i$ with an associated weight $w_i$ such that $P(\hat{F}_j) = p$ for some probability $p$ and a ground instance $\hat{F}_j$ of $F_i$ – assuming, for the time being, that $F_i$ is the only formula in the network. If we set $w_i = \log(p_1/p_2)$, then, for $x_1 \models \hat{F}_j$ and $x_2 \not\models \hat{F}_j$, we have that $\omega(x_1) : \omega(x_2) = p_1/p_2 : 1 = p_1 : p_2$ (if $x_1$ and $x_2$ are equivalent as far as the truth values of other ground instances of $F_i$ are concerned). To compute the probability $P(\hat{F}_j)$, we need to consider $s(\hat{F}_j)$ and $s(\neg\hat{F}_j)$. With $\#_{\hat{F}_j}$ as the number of models of the ground formula $\hat{F}_j$, it follows that if $\#_{\hat{F}_j} = \#_{\neg\hat{F}_j}$, then $s(\hat{F}_j) : s(\neg\hat{F}_j) = p_1 : p_2$ and therefore $P(\hat{F}_j) = p_1/(p_1 + p_2)$. A particularly relevant case where $\#_{\hat{F}_j} = \#_{\neg\hat{F}_j}$ holds is the case where $F_i$ is an atomic formula such as $\mathit{bird}(e)$. Thus, the probability $P(\mathit{bird}(e))$ can be set to $p$ for all entities $e$ using

$$\log(p/(1-p)) \quad \mathit{bird}(e).$$

If, however, the weight $w_i$ applies asymmetrically to the set of possible worlds, i.e. if $\#_{\hat{F}_j} \neq \#_{\neg\hat{F}_j}$, then this has to be taken into consideration. If, for example, the formula $\hat{F}_j$ was satisfied in the majority of possible worlds, then using $\log(p/(1-p))$ as a weight would result in $P(\hat{F}_j)$ being larger than $p$. To set $P(\hat{F}_j) = p$, we

need to find a factor $\hat{\lambda}_j$ such that

$$P(\hat{F}_j) = \frac{s(\hat{F}_j)}{s(\hat{F}_j) + s(\neg\hat{F}_j)} = \frac{\#_{\hat{F}_j}\hat{\lambda}_j}{\#_{\hat{F}_j}\hat{\lambda}_j + \#_{\neg\hat{F}_j}1} = p$$
$$\Rightarrow \hat{\lambda}_j = \#_{\neg\hat{F}_j}p/(\#_{\hat{F}_j}(1-p)) \tag{5}$$

For instance, if $F_i$ is an implication such as $\mathit{bird}(e) \Rightarrow \mathit{flies}(e)$, then we can achieve $P(\mathit{bird}(e) \Rightarrow \mathit{flies}(e)) = p$ for all $e$ using

$$\log(p/(3-3p)) \quad \mathit{bird}(e) \Rightarrow \mathit{flies}(e)$$

since $\#_{\hat{F}_j} : \#_{\neg\hat{F}_j} = 3:1$ in this case.

In all the above considerations, we have made the assumption that the sets of ground atoms appearing in any two ground instances of $F_i$ is disjoint, for if they were to overlap, then there would be interactions between the formulas that might already be too difficult to manually foresee.

**Weights of mutually exclusive formulas.** There is one simple case where we can easily cope with ground atoms appearing in more than one ground formula – the case where the formulas are mutually exclusive, such that in any possible world, at most one of the formulas is true. For instance, if we have both the formula $\mathit{bird}(e)$ and the formula $\neg\mathit{bird}(e)$, then we can achieve $P(\mathit{bird}(e)) = p$ using

$$\begin{array}{ll} \log(p) & \mathit{bird}(e) \\ \log(1-p) & \neg\mathit{bird}(e). \end{array}$$

Because each of the cases is considered in an explicit factor, we need not use odds to establish the desired ratio; the above weighting will immediately ensure that $s(\hat{F}_j) : s(\neg\hat{F}_j) = p : (1-p)$ and therefore $P(\hat{F}_j) = p$ for all instances $\hat{F}_j$ of $\mathit{bird}(e)$.

This procedure straightforwardly translates to larger sets of mutually exclusive formulas. If, for instance, we have a predicate declared as $\mathit{isa}(\mathit{entity}, \mathit{type}!)$, i.e. an entity belongs to precisely one of several types, then if the set of types is e.g. $\{\mathit{Mammal}, \mathit{Bird}, \mathit{Reptile}\}$ and the corresponding probabilities are $p_1, p_2, p_3$, we could set:

$$\begin{array}{ll} \log(p_1) & \mathit{isa}(e, \mathit{Mammal}) \\ \log(p_2) & \mathit{isa}(e, \mathit{Bird}) \\ \log(p_3) & \mathit{isa}(e, \mathit{Reptile}) \end{array}$$

It is desirable for the set of formulas whose weights we set in this way to not only be mutually exclusive but also exhaustive – i.e. exactly one of the formulas should be true for every given world and every binding of the variables – because cases not mentioned in our set of formulas obtain an implicit weight of 0 by default, which in turn implies a higher probability for cases not mentioned, since $0 \geq \log(p_i)$.

It is also important to note that for a mutually exclusive and exhaustive set of formulas, weights are meaningful *only* relative to each other. In particular, for a probability value $p_i \in [0;1[$ the corresponding weight $\log(p_i)$ is smaller than

0, but this does not at all imply that the corresponding formula being true will lower the probability of worlds satisfying it. In fact, we can add arbitrary offsets $\delta \in \mathbb{R}$ to all of the formula weights without affecting the distribution $P(X = x)$, because if the formulas are mutually exclusive and exhaustive, then the offset will apply to each possible world in exactly the same way: If there are $k$ ground instances of the mutually exclusive and exhaustive set of formulas, then $\delta$ is summed $k$ times for each possible world $x \in \mathcal{X}$ and we thus obtain:

$$P(x) = \frac{\exp\left(\sum_j \hat{w}_j \hat{f}_j(x) + k\delta\right)}{\sum_{x' \in \mathcal{X}} \exp\left(\sum_j \hat{w}_j \hat{f}_j(x') + k\delta\right)} = \frac{\exp\left(\sum_j \hat{w}_j \hat{f}_j(x)\right)}{\sum_{x' \in \mathcal{X}} \exp\left(\sum_j \hat{w}_j \hat{f}_j(x')\right)} \cdot \frac{\cancel{\exp(k\delta)}}{\cancel{\exp(k\delta)}} \quad (6)$$

## 4 The Probabilistic Implication Fallacy

A causal model is often an intuitive representation of a generative stochastic process. Causality dictates a temporal ordering in which values are assigned to variables: If A is the cause of B, then we first determine whether or not A is the case, and depending on the result, we determine whether B is the also case. In the strictly logical case, we can use the implication $A \Rightarrow B$ to represent logical causal influence. In the probabilistic case, the influence we need to model is "If A holds, then B holds with some probability", which corresponds to the conditional probability $P(B \mid A)$. The *probabilistic implication fallacy* lies in the assumption that a softened version of the implication $A \Rightarrow B$ is an adequate representation for probabilistic causal influence.

**Fallacy 1: Probabilistic implications are equivalent to conditional probabilities.** Most likely, one of the reasons for people to think of implications and conditional probabilities as being inextricably linked is that in the strictly logical case, there is indeed an obvious connection. Consider the classical example of modelling the probability with which birds are able to fly. If all birds are able to fly, $P(\textit{flies} \mid \textit{bird}) = 1$. In this case, the weighted formula

(hard) $\quad \textit{bird}(e) \Rightarrow \textit{flies}(e),$

which, by definition, implies $P(\textit{bird} \Rightarrow \textit{flies}) = 1$, has related semantics. If *bird* is true, we recover *flies* with probability 1.

In general, however, there is no immediate correspondence between the probability of the implication and the conditional probability. If we set the probability $P(\textit{bird}(e) \Rightarrow \textit{flies}(e))$ to $p \in ]0; 1]$, which we can achieve by setting the weight of the implication to $\log(p/(3-3p))$ (Eq. 5) or, equivalently, by using the more explicit group of mutually exclusive formulas

$$\begin{array}{ll}
\log(p/3) & \textit{flies}(e) \wedge \quad \textit{bird}(e) \\
\log(p/3) & \textit{flies}(e) \wedge \neg \textit{bird}(e) \\
\log(p/3) & \neg \textit{flies}(e) \wedge \neg \textit{bird}(e) \\
\log(1-p) & \neg \textit{flies}(e) \wedge \quad \textit{bird}(e)
\end{array}$$

then we obtain $c := P(\textit{flies}(e) \mid \textit{bird}(e)) = (p/3)/(p/3+1-p) = p/(3-2p)$. This expression is equal to $p$ only for $p = 1$ and $p = 0.5$. For example, if $p = 0.7$, then the conditional probability is actually 0.4375.

Of course, it is possible to set the probability $p$ of the implication such that it will result in any given conditional probability for $c$ (specifically, $p/(3-2p) = c \Leftrightarrow p = 3c/(1+2c)$). However, as is evident from the above reformulation, the probabilistic implication has bearing not only on the conditional probability of *flies* given *bird* but also on the marginal distributions of both atoms. For instance, for $p = 1$ and $p = 0.7$, we obtain $P(\mathit{bird}(e)) = 1/3 = 0.\overline{3}$ and $P(\mathit{bird}(e)) = 7/15 = 0.4\overline{6}$ respectively. From the equivalence $\mathit{bird}(e) \Rightarrow \mathit{flies}(e) \equiv \neg\mathit{bird}(e) \vee \mathit{flies}(e)$, it should be clear that a greater probability of $\mathit{bird}(e) \Rightarrow \mathit{flies}(e)$ will result in a lower probability of $\mathit{bird}(e)$.

Therefore, a probabilistic implication is clearly inappropriate for the representation of a conditional probability.[4] The probability/weight of an implication does not correspond to a conditional probability in the general case, and implications have influence on the distribution beyond the conditional probability that was intended to be affected.

**Fallacy 2: A single implication is sufficient for the representation of probabilistic causal influence.** Motivated by the top-down view (V1), it might seem natural to replace the logical influence of *bird* on *flies*, which, in a logical model, might have been represented using the implication $\forall e.\ \mathit{bird}(e) \Rightarrow \mathit{flies}(e)$, with a weighted version of the same formula in an MLN. In other words, it might seem natural to make the transition from logical to probabilistic knowledge by simply adding weights to the formulas that are not universally true and thereby make room for exceptions. Unfortunately, this rather intriguing concept turns out to be insufficient if formerly logical dependencies are to be replaced by probabilistic dependencies.

In the strictly logical case, we want to conclude *flies* from *bird*, i.e. if $\mathit{bird}(e)$ holds, we expect to recover $\mathit{flies}(e)$ with probability 1. When we make the transition to the probabilistic realm, a causal model will need to represent the degree to which we can conclude *flies* from *bird* and thus represent the conditional probability $P(\mathit{flies}(e) \mid \mathit{bird}(e))$. As we saw above, the probabilistic implication is a poor candidate for the representation of this probability.

To determine how to best represent such conditional probabilities, a simple analysis suffices. By definition, $P(\mathit{flies}(e) \mid \mathit{bird}(e)) \propto P(\mathit{flies}(e) \wedge \mathit{bird}(e))$. The conditional probability is thus concerned with defining the degree to which *flies* and *bird* co-occur. If $\mathit{bird}(e)$ is given as evidence, we need not concern ourselves with how $P(\mathit{bird}(e))$ may be modelled, and by attaching a weight to $\mathit{flies}(e) \wedge \mathit{bird}(e)$, we can establish any ratio between $\mathit{flies}(e) \wedge \mathit{bird}(e)$ and $\neg\mathit{flies}(e) \wedge \mathit{bird}(e)$. In particular, the ratio $p : 1-p$ can be established using

$$\log(p/(1-p)) \quad \mathit{flies}(e) \wedge \mathit{bird}(e),$$

setting the conditional probability to $p$. As long as $\mathit{bird}(e)$ is given, we could, notably, have used the formula $\mathit{flies}(e) \Rightarrow \mathit{bird}(e)$ to the same effect. Unfortunately, neither formulation leaves the marginal probability of $\mathit{bird}(e)$ unchanged,

[4] We are, of course, not the first to make this observation, as it essentially follows immediately from the definition of the implication. Joseph Y. Halpern has, for instance, remarked on an analogous issue in his analysis of first-order logics of probability [3].

because both affect cases where $bird(e)$ holds and cases where $\neg bird(e)$ holds asymmetrically. The addition of a causal influence (via conditional probabilities) should, however, have no impact on the marginal probability of the cause. Therefore, we need to find weighted formulas that will represent the conditional probability we want to represent whilst guaranteeing that $P(bird(e))$ and therefore the ratio $s(bird(e)) : s(\neg bird(e))$ (for an arbitrary but fixed $e$) remains unaffected. If we consider all the configurations of the clique connecting $bird(e)$ and $flies(e)$ explicitly, we will have a maximum of flexibility to satisfy these restrictions.

$$\begin{array}{lrl} w_1 = \log(\lambda_1) & flies(e) \wedge & bird(e) \\ w_2 = \log(\lambda_2) & \neg flies(e) \wedge & bird(e) \\ w_3 = \log(\lambda_3) & flies(e) \wedge & \neg bird(e) \\ w_4 = \log(\lambda_4) & \neg flies(e) \wedge & \neg bird(e) \end{array}$$

In order to represent the conditional probability $P(flies(e) \mid bird(e)) = p$ and thus establish the ratio between $s(flies(e) \wedge bird(e))$ and $s(\neg flies(e) \wedge bird(e))$ as $p : 1 - p$, we can simply set $\lambda_1 = p$ and $\lambda_2 = 1 - p$. If, prior to the addition of the four above formulas to our MLN, the model contains no information about the distribution of $flies(e)$ and therefore $S_B := s(flies(e) \wedge bird(e)) = s(\neg flies(e) \wedge bird(e))$ and $S_{\neg B} := s(flies(e) \wedge \neg bird(e)) = s(\neg flies(e) \wedge \neg bird(e))$, then $s(bird(e)) : s(\neg bird(e))$ will remain unaffected if $\lambda_3 + \lambda_4 = \lambda_1 + \lambda 2$, since

$$\frac{s(bird(e))}{s(\neg bird(e))} = \frac{2 \cdot S_B}{2 \cdot S_{\neg B}} \stackrel{!}{=} \frac{S_B\lambda_1 + S_B\lambda_2}{S_{\neg B}\lambda_3 + S_{\neg B}\lambda_4} \Leftrightarrow \frac{\lambda_1 + \lambda_2}{\lambda_3 + \lambda 4} = 1. \tag{7}$$

The use of complementary probability values for $\lambda_3$ and $\lambda_4$ trivially satisfies the equation, and the four formulas above will thus collectively represent the entire conditional distribution of *flies* given *bird*. Therefore, if unwanted side-effects are to be avoided, the representation of a full conditional distribution is perhaps the most obvious solution.

We can represent a direct factorisation of the full joint, i.e. $P(bird(e), flies(e)) = P(flies(e) \mid bird(e))P(bird(e))$, by adding the marginal probability $P(bird(e)) = p_b$ to the model using either $\langle bird(e), \log(p_b/(1-p_b))\rangle$ or the mutually exclusive pair of $bird(e)$ and $\neg bird(e)$ with weights $\log(p_b)$ and $\log(1-p_b)$. In the latter case, we obtain a model that is completely analogous to a directed model in the sense that it represents precisely the same factors with $Z = 1$.

We have now established that, in order to model causal influence, the use of implications is certainly not as helpful as it may, at first, have appeared. Indeed, the semantics of implications do not seem to translate well to the probabilistic setting. Nevertheless, it is worth noting that it would have been possible to use implications to achieve precisely the same effect as the four conjunctions above. Specifically, the following two transformations are semantically equivalent:

$$\begin{array}{l|ll} -w_1 & (w_2 + w_3 + w_4 - 2w_1)/3 & flies(e) \Rightarrow \neg bird(e) \\ -w_2 & (w_1 + w_3 + w_4 - 2w_2)/3 & \neg flies(e) \Rightarrow \neg bird(e) \\ -w_3 & (w_1 + w_2 + w_4 - 2w_3)/3 & flies(e) \Rightarrow bird(e) \\ -w_4 & (w_1 + w_2 + w_3 - 2w_4)/3 & \neg flies(e) \Rightarrow bird(e) \end{array}$$

The second transformation even retains the normalisation constant. However, we cannot think of a good reason to prefer them – to the contrary: Since implications are not mutually exclusive, their impact on possible worlds is considerably more difficult to interpret.

**Syntactic Sugar.** From the above considerations, we can draw an important conclusion: Simply adding weights to the logical formulas we might have used in the non-probabilistic case is unlikely to be sufficient if probabilistic dependencies are to be captured to their full extent. While Markov logic networks *are* a generalisation of logic that soundly integrates probabilistic semantics, thinking in logical terms is, for the most part, a hindrance as far as the probabilistic knowledge engineering process is concerned. In fact, we suggest that logical formulations involving (bi)implications be used primarily for the purpose of specifying hard constraints. We argue that, with regard to probabilistic aspects, formulas are mere syntactic sugar for the specification of a subset of a clique's configurations, and whether or not the logical form fosters comprehensibility of the effect of a weighted formula is debatable. In particular, we discourage the use of implications in probabilistic contexts because implications are typically associated with a directedness which, in probabilistic settings, may not reflect the expected semantics. We argue that the probabilistic counterpart of an implication is not a probabilistic implication but rather a conditional probability (distribution). For the representation of non-deterministic features, an implication such as as $\mathit{bird}(e) \Rightarrow \mathit{flies}(e)$ should be used only if the negation $\mathit{bird}(e) \wedge \neg\mathit{flies}(e)$ is deemed equally appropriate for the task (where the negated form is perhaps the more readable alternative). The key question to ask is: Is the set of partial configurations affected by an implication truly the one we seek to affect or is it the co-occurrence of antecedent and consequent (as in a conditional probability statement) we are interested in?

## 5 Shallow Transfer

As mentioned in the introduction, one of the key ideas of first-order probabilistic languages such as MLNs is to represent models that generalise across domains – i.e. the principle of *shallow transfer*. For any particular domain (set of entities), we, ideally, want the model to generate a specific ground model that is "adequate". In this section, we discuss cases where the transition from one domain to another can affect the distribution in perhaps unexpected ways. In short, *the number of entities may matter significantly.*

**Invariant properties of distributions across domains.** In many scenarios, it is reasonable to assume that the probability distributions represented by a statistical relational model will possess certain invariants that will hold regardless of the instantiation. For instance, in the introductory restaurant example, it might, for instance, be reasonable to expect that the probability of being female should be 0.5 for all people about whom we do not have any evidence. Indeed, if we assume that the model is to represent a (temporally ordered) generative stochastic process which first generates people along with their gender and subsequently generates further properties and links to other entities depending on the gender,

then the probability of being female must remain constant in the absence of evidence. Similarly, an invariant might exist for the probability of a dish being vegetarian.

Consider this simplified version of the introductory example,

| | |
|---|---|
| $\log(0.25/0.75)$ | $\mathit{vegetarianDish}(d)$ |
| $\log(0.15/0.85)$ | $\mathit{vegetarian}(p)$ |
| (hard) | $\mathit{orders}(p, d) \wedge \mathit{vegetarian}(p) \Rightarrow \mathit{vegetarianDish}(d)$ |

where the *orders* predicate is declared as functional ($\mathit{orders}(\mathit{person}, \mathit{dish}!)$, see footnote 2). As we already know from Section 4, the third formula will have influence on the marginal probability of a dish being vegetarian. Therefore, it is not surprising to learn the probability will be greater than the probability 0.25 that is indicated by the first formula. Perhaps more surprising is the fact that the influence of the hard formula varies with the number of entities in the domain. For example, the marginal probability of a given dish being vegetarian is approximately 0.28 for a domain where there are two people and 0.39 for a domain where there are four people and one dish.

The variations are due to combinatorial effects [5]. Without the third formula, if there are $N_P$ persons and $N_D$ dishes, then there are $\alpha(N_P, N_D) := 2^{N_P+N_D} \cdot {N_D}^{N_P}$ possible worlds with non-zero probability (given that every person consumes exactly one dish). For any pair of possible worlds $\langle x, x' \rangle$ where $x$ and $x'$ differ only in the truth value of $\mathit{vegetarianDish}(d)$ for some dish $d$ ($x \models \mathit{vegetarianDish}(d)$), $\omega(x) : \omega(x') = 25 : 75$. The inclusion of the hard constraint causes the probability of $\delta(N_P, N_D) := \sum_{v=1}^{N_P} \binom{N_P}{v} \sum_{m=1}^{v} \binom{v}{m} \sum_{i=1}^{N_D} \binom{N_D}{i} i^m (N_D - i)^{v-m} {N_D}^{N_P-v}$ worlds violating the constraint to be set to zero ($v$ is the number of vegetarians, $m$ is the number of meat-eating vegetarian, and $i$ is the number of non-vegetarian dishes), such that for some pairs $\langle x, x' \rangle$, the ratio $\omega(x) : \omega(x')$ becomes $25 : 0$, thus causing worlds with vegetarian dishes to become more probable. Since the fraction $\delta(N_P, N_D)/\alpha(N_P, N_D)$ increases as either $N_P$ or $N_D$ increases, $P(\mathit{vegetarianDish}(d))$ varies with the number of people and dishes accordingly.[5]

There is, unfortunately, no way of adjusting the parameters of the model such that it will invariantly compute a fixed marginal probability for *vegetarianDish*. Any adjustments made to the weight of the first formula will correct the probability only for a given number of entities.

Moreover, even if we replace the third constraint with the full conditional distribution of *orders* given *vegetarian* and *vegetarianDish* (which would, by definition, leave the marginals unchanged as explained in Section 4), there will still be interactions with domain size owing to the constraint that requires exactly one dish to be consumed by each person. Therefore, in conclusion, there are probabilistic invariants that simply cannot be represented. Because the notion of invariants that are to hold in all instantiations is all but far-fetched (any model of a stochastic process that is causally ordered will typically exhibit invariants),

[5] Note that there is an analogous effect for the predicate *vegetarian*. For reasons of brevity, we leave its analysis to the reader.

it is a key issue of knowledge engineering, and we deem it essential to be aware of the ramifications.

Several concepts have been introduced to address such issues. For instance, MLNs can be augmented with constraints on formula probabilities, which, in an online adaptation step, can be used to compute corresponding formula weights that will satisfy these constraints [5]. In [1], the notion is extended to constraints on conditional probabilities. Because a computationally expensive online adjustment of parameters may be infeasible, adaptive Markov logic networks were proposed in order to allow weights to be directly represented as functions of domain-specific parameters, which can be learnt from multiple training databases encompassing domains of variable size [4].

**Parameter learning may not guarantee that shallow transfer will work as expected.** From the analyses above, it should be clear that the manual engineering of MLNs is inherently difficult. As soon as we depart from the pattern of modelling marginal and conditional distributions using mutually exclusive and exhaustive formulas, the interactions between formulas will typically be impossible to predict intuitively. Learning parameters from data (or even the structure of the model) is often advisable. However, it is important to realise that learning may not guarantee that shallow transfer will work as expected. Consider the following MLN,

| | | |
|---|---|---|
| $w_1$ | $w_1 + \delta$ | $vegetarian(p)$ |
| $w_2$ | $w_2$ | $vegetarianDish(d)$ |
| $w_3$ | $w_3 - \delta/N_D$ | $orders(p,d) \wedge\ vegetarian(p) \wedge\ vegetarianDish(d)$ |
| $w_4$ | $w_4 - \delta/N_D$ | $\neg orders(p,d) \wedge\ vegetarian(p) \wedge\ vegetarianDish(d)$ |
| $w_5$ | $w_5 - \delta/N_D$ | $orders(p,d) \wedge\ vegetarian(p) \wedge \neg vegetarianDish(d)$ |
| $w_6$ | $w_6 - \delta/N_D$ | $\neg orders(p,d) \wedge\ vegetarian(p) \wedge \neg vegetarianDish(d)$ |
| $w_7$ | $w_7$ | $orders(p,d) \wedge \neg vegetarian(p) \wedge\ vegetarianDish(d)$ |
| $w_8$ | $w_8$ | $\neg orders(p,d) \wedge \neg vegetarian(p) \wedge\ vegetarianDish(d)$ |
| $w_9$ | $w_9$ | $orders(p,d) \wedge \neg vegetarian(p) \wedge \neg vegetarianDish(d)$ |
| $w_{10}$ | $w_{10}$ | $\neg orders(p,d) \wedge \neg vegetarian(p) \wedge \neg vegetarianDish(d)$ |

which is capable of representing a factorisation analogous to a directed model: The first two formulas can represent the marginal distributions of *vegetarian* and *vegetarianDish*, and the remaining formulas can represent the four conditional distributions of *orders* given *vegetarian* and *vegetarianDish*. For a case where there are no constraints on the *orders* relation (such as the requirement that exactly one dish must be consumed by everyone), this set of of formulas will allow us to represent invariant marginals for both *vegetarian* and *vegetarianDish* and, by assigning a hard negative weight to the fifth formula and a weight of $0 = \log(1.0)$ to the sixth, it also allows us to represent the "vegetarian constraint".

However, when learning the weights from data, we will not necessarily recover the invariants – regardless of the degree to which the empirical frequencies in the data match the true parameters (and regardless of the number of independent training databases we may use). This is due to the fact that the learning problem is often ill-posed [5]. For the above MLN, there is an infinite number of weight vectors that represent precisely the same distribution – for a given number of

entities. In particular, if the number of dishes is $N_D$, then for any $\delta \in \mathcal{R}$, the two weight vectors listed above are equivalent and represent precisely the same distribution over possible worlds. As soon as the number of dishes changes, however, the distributions will differ significantly [5]. In our example, the marginal probability of being a vegetarian will vary widely.

A maximum likelihood learner will have no preference of one weight vector over the other. By applying a form of regularisation (e.g. using Gaussian 0-mean priors on the weights as proposed in [7]), we ensure that the optimisation process returns a unique solution. However, that solution is not necessarily the one we expect. Indeed, the solution that results can, as far as generalisation across domains is concerned, be regarded as arbitrary. Hence the soundness of shallow transfer cannot be taken for granted. Indeed, introducing additional knowledge in the form of constraints is necessary in order for the learning process to determine weights that will ultimately represent what is intended. From a knowledge engineering perspective, this is an important point, for it suggests that the task of knowledge engineering in MLNs may require considerations that go beyond the selection of the "right" formulas even when the parameters are learnt from data.

## 6 Conclusion

In this paper, we scrutinised a number of knowledge engineering questions that arise in Markov logic networks (MLNs). While our collection is by no means complete, we have, nevertheless, addressed many issues that are of high practical relevance. We described the fundamental semantics of MLNs and explained the considerations that are necessary for the representation of simple probabilistic properties with MLNs. With the probabilistic implication fallacy, we explicitly treated one of the most common sources of error, explaining the inherent predicament in softening logic as well as its resolution. Finally, we discussed and summarised conditions under which generalisation across domains can be problematic even for cases where models are learnt from data.

## References


1. J. Fisseler. Toward Markov Logic with Conditional Probabilities. In *FLAIRS Conference*, pages 643–648, 2008.
2. L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2007.
3. J. Y. Halpern. An Analysis of First-Order Logics of Probability. *Artificial Intelligence*, 46:311–350, 1990.
4. D. Jain, A. Barthels, and M. Beetz. Adaptive Markov Logic Networks: Learning Statistical Relational Models with Dynamic Parameters. In *19th European Conference on Artificial Intelligence (ECAI)*, pages 937–942, 2010.
5. D. Jain, B. Kirchlechner, and M. Beetz. Extending Markov Logic to Model Probability Distributions in Relational Domains. In *30th German Conference on Artificial Intelligence (KI-2007)*, pages 129–143, 2007.
6. D. Lowd and P. Domingos. Recursive Random Fields. In *20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 950–955, 2007.
7. M. Richardson and P. Domingos. Markov Logic Networks. *Mach. Learn.*, 62(1-2):107–136, 2006.

# Analyzing Inconsistencies in Probabilistic Conditional Knowledge Bases using Continuous Inconsistency Measures

Matthias Thimm

Technische Universität Dortmund, Germany

**Abstract.** Probabilistic conditional logic is a knowledge representation formalism that uses probabilistic conditionals (if-then rules) to model uncertain and incomplete information. By applying the principle of maximum entropy one can reason with a set of probabilistic conditionals in an information-theoretical optimal way, provided that the set is consistent. As in other fields of knowledge representation, consistency of probabilistic conditional knowledge bases is hard to ensure if their size increases or multiple sources contribute pieces of information. In this paper, we discuss the problem of analyzing and measuring inconsistencies in probabilistic conditional logic by investigating inconsistency measures that support the knowledge engineer in maintaining a consistent knowledge base. An inconsistency measure assigns a numerical value to the severity of an inconsistency and can be used for restoring consistency. Previous works on measuring inconsistency consider only qualitative logics and are not apt for quantitative logics because they assess severity of inconsistency without considering the probabilities of conditionals. Here, we investigate *continuous* inconsistency measures which allow for a more fine-grained and continuous measurement.

## 1 Introduction

Inconsistencies arise easily when experts share their beliefs in order to build a joint knowledge base. Although these inconsistencies often affect only a little portion of the knowledge base or emerge from only little differences in the experts' beliefs, they cause severe damage. In particular, for knowledge bases that use classical logic for knowledge representation, inconsistencies render the whole knowledge base useless, due to the well-known principle *ex falso quodlibet*. Therefore reasoning under inconsistency is an important field in knowledge representation and reasoning and there are basically two paradigms for approaching this issue. On the one hand one can live with inconsistencies and develop reasoning mechanisms that allow for consistent inference in the presence of inconsistent information, cf. e. g. *paraconsistent* and *default logics* [8]. On the other hand one can rely on classical inference mechanisms and ensure that knowledge bases are consistent, cf. e. g. approaches to *belief revision* and *information fusion* [2]. In this paper we employ probabilistic conditional logic [5] for knowledge representation. The basic notion of probabilistic conditional logic is that of a *probabilistic*

*conditional* which has the form $(\psi \mid \phi)[d]$ with the commonsense meaning "if $\phi$ is true then $\psi$ is true with probability $d$". A popular choice for reasoning with sets of probabilistic conditionals is model-based inductive reasoning based on the principle of maximum entropy [6, 5]. However, a prerequisite for applying this principle is the consistency of the set, i. e. the existence of at least one probability function that satisfies all probabilistic conditionals.

In this paper we investigate the issue of inconsistency in probabilistic conditional logic from an analytical perspective. One way to analyze inconsistencies is by measuring them. For the framework of classical logic, several approaches to analyze and measure inconsistency have been proposed—see e. g. [3]—and it is straightforward to apply those measures to the framework of probabilistic conditional logic [11]. However, those approaches do not grasp the nuances of probabilistic knowledge and allow only for a very coarse assessment of the severity of inconsistencies. In particular, those approaches do not take the crucial role of probabilities into account and exhibit a discontinuous behavior in measuring inconsistency. That is, a slight modification of the probability of a conditional in a knowledge base may yield a discontinuous change in the value of the inconsistency. In this paper, we consider measuring inconsistency in probabilistic conditional logic and continue previous work [10] in several aspects. First, we propose several novel principles for inconsistency measurement. Second, we pick up an extended logical formalization [7] of the inconsistency measure proposed in [10] and define a family of inconsistency measures based on minimizing the $p$-norm distance of a knowledge base to consistency. Third, we propose a novel compound measure that solves an issue with the previous measure and investigate its properties.

The rest of this paper is organized as follows. In Sec. 2 we give a brief overview on probabilistic conditional logic and continue in Sec. 3 with presenting a set of rationality postulates for continuous inconsistency measurement. We propose a family of inconsistency measures and a compound measure in Sec. 4 and analyze their properties in Sec. 5. We briefly review related work in Sec. 6 and conclude with a summary in Sec. 7.

## 2 Probabilistic Conditional Logic

Let $\mathsf{At}$ be a propositional signature, i. e. a finite set of propositional atoms. Let $\mathcal{L}(\mathsf{At})$ be the corresponding propositional language generated by the atoms in $\mathsf{At}$ and the connectives $\wedge$ (*and*), $\vee$ (*or*), and $\neg$ (*negation*). For $\phi, \psi \in \mathcal{L}(\mathsf{At})$ we abbreviate $\phi \wedge \psi$ by $\phi\psi$ and $\neg\phi$ by $\overline{\phi}$. The symbols $\top$ and $\bot$ denote *tautology* and *contradiction*, respectively. We use *possible worlds*, i. e. syntactical representations of *truth assignments*, for interpreting sentences in $\mathcal{L}(\mathsf{At})$. A possible world $\omega$ is a complete conjunction, i. e. a conjunction that contains for each $a \in \mathsf{At}$ either $a$ or $\neg a$. Let $\Omega(\mathsf{At})$ denote the set of all possible worlds. A possible world $\omega \in \Omega(\mathsf{At})$ *satisfies* an atom $a \in \mathsf{At}$, denoted by $\omega \models a$ if and only if $a$ positively appears in $\omega$. The entailment relation $\models$ is extended to arbitrary formulas in

$\mathcal{L}(\mathsf{At})$ in the usual way. Formulas $\psi, \phi \in \mathcal{L}(\mathsf{At})$ are *equivalent*, denoted by $\phi \equiv \psi$, if and only if $\omega \models \phi$ whenever $\omega \models \psi$ for every $\omega \in \Omega(\mathsf{At})$.

The central notion of probabilistic conditional logic [5] is that of a *probabilistic conditional*.

**Definition 1.** *If $\phi, \psi \in \mathcal{L}(\mathsf{At})$ with $d \in [0,1]$ then $(\psi \,|\, \phi)[d]$ is called a* probabilistic conditional.

A probabilistic conditional $c = (\psi \,|\, \phi)[d]$ is meant to describe a probabilistic *if-then* rule, i. e., the informal interpretation of $c$ is that "If $\phi$ is true then $\psi$ is true with probability $d$" (see below). If $\phi \equiv \top$ we abbreviate $(\psi \,|\, \phi)[d]$ by $(\psi)[d]$. Further, for $c = (\psi \,|\, \phi)[d]$ we denote with $\mathsf{head}(c) = \psi$ the *head* of $c$, with $\mathsf{body}(c) = \phi$ the *body* of $c$, and with $\mathsf{prob}(c) = d$ the *probability* of $c$. Let $\mathcal{C}(\mathcal{L}(\mathsf{At}))$ denote the set of all probabilistic conditionals with respect to $\mathcal{L}(\mathsf{At})$.

**Definition 2.** *A* knowledge base $\mathcal{K}$ *is an ordered finite multi-subset of $\mathcal{C}(\mathcal{L}(\mathsf{At}))$, i. e. it holds that $\mathcal{K} = \langle c_1, \ldots, c_n \rangle$ for some $c_1, \ldots, c_n \in \mathcal{C}(\mathcal{L}(\mathsf{At}))$.*

We impose an ordering on the conditionals in a knowledge base $\mathcal{K}$ only for technical convenience. The order can be arbitrary and has no further meaning other than to enumerate the conditionals of a knowledge base in an unambiguous way. For similar reasons we allow a knowledge base to contain the same probabilistic conditional more than once. We come back to the reasons for these design choices later. For knowledge bases $\mathcal{K} = \langle c_1, \ldots, c_n \rangle$, $\mathcal{K}' = \langle c'_1, \ldots, c'_m \rangle$ and a probabilistic conditional $c$ we define $c \in \mathcal{K}$ via $c \in \{c_1, \ldots, c_n\}$, $\mathcal{K} \subseteq \mathcal{K}'$ via $\{c_1, \ldots, c_n\} \subseteq \{c'_1, \ldots, c'_m\}$, and $\mathcal{K} = \mathcal{K}'$ via $\{c_1, \ldots, c_n\} = \{c'_1, \ldots, c'_m\}$. The union of belief bases is defined via concatenation.

Semantics are given to probabilistic conditionals by *probability functions* on $\Omega(\mathsf{At})$. Let $\mathcal{F}$ denote the set of all probability functions $P : \Omega(\mathsf{At}) \to [0,1]$. A probability function $P \in \mathcal{F}$ is extended to formulas $\phi \in \mathcal{L}(\mathsf{At})$ via

$$P(\phi) = \sum_{\omega \in \Omega(\mathsf{At}), \omega \models \phi} P(\omega) \quad .$$

If $P \in \mathcal{F}$ then $P$ satisfies a probabilistic conditional $(\psi \,|\, \phi)[d]$, denoted by $P \models^{pr} (\psi \,|\, \phi)[d]$, if and only if $P(\psi\phi) = dP(\phi)$. Note that we do not define probabilistic satisfaction via $P(\psi \,|\, \phi) = {}^{P(\psi\phi)}/_{P(\phi)} = d$ in order to avoid a case differentiation for $P(\phi) = 0$, cf. [6]. A probability function $P$ satisfies a knowledge base $\mathcal{K}$ (or is a model of $\mathcal{K}$), denoted by $P \models^{pr} \mathcal{K}$, if and only if $P \models^{pr} c$ for every $c \in \mathcal{K}$. Let $\mathsf{Mod}(\mathcal{K})$ be the set of models of $\mathcal{K}$. If $\mathsf{Mod}(\mathcal{K}) = \emptyset$ then $\mathcal{K}$ is *inconsistent*.

*Example 1.* Consider the knowledge base

$$\mathcal{K} = \langle (f \,|\, b)[0.9], (b \,|\, p)[1], (f \,|\, p)[0.1] \rangle$$

with the intuitive meaning that birds ($b$) usually (with probability 0.9) fly ($f$), that penguins ($p$) are always birds, and that penguins usually do not fly (only

with probability 0.1). The knowledge base $\mathcal{K}$ is consistent as for e. g. $P \in \mathcal{F}$ with

$$P(bfp) = 0.005 \qquad P(bf\overline{p}) = 0.49 \qquad P(b\overline{f}p) = 0.045 \qquad P(b\overline{f}\overline{p}) = 0.01$$
$$P(\overline{b}fp) = 0.0 \qquad P(\overline{b}f\overline{p}) = 0.2 \qquad P(\overline{b}\overline{f}p) = 0.0 \qquad P(\overline{b}\overline{f}\overline{p}) = 0.25$$

it holds that $P \models^{pr} \mathcal{K}$ as e. g. $P(b) = P(bfp) + P(bf\overline{p}) + P(b\overline{f}p) + P(b\overline{f}\overline{p}) = 0.55$ and $P(bf) = P(bfp) + P(bf\overline{p}) = 0.495$ and therefore $P(f \mid b) = {}^{P(bf)}/_{P(b)} = 0.9$.

A probabilistic conditional $(\psi \mid \phi)[d]$ is *normal* if and only if there are $\omega, \omega' \in \Omega(\mathsf{At})$ with $\omega \models \psi\phi$ and $\omega' \models \overline{\psi}\phi$.[1] In other words, a probabilistic conditional $c$ is normal if it is satisfiable but not tautological.

*Example 2.* The probabilistic conditionals $c_1 = (\top \mid a)[1]$ and $c_2 = (\overline{a} \mid a)[0.1]$ are *not* normal as $c_1$ is tautological (there is no $\omega \in \Omega(\mathsf{At})$ with $\omega \models \overline{\top}a$ as $\overline{\top}a \equiv \bot$) and $c_2$ is not satisfiable (there is no $\omega \in \Omega(\mathsf{At})$ with $\omega \models \overline{a}a$ as $\overline{a}a \equiv \bot$)

As a technical convenience, for the rest of this paper we consider only normal probabilistic conditionals, so let $\mathbb{K}$ be the set of all knowledge bases of $\mathcal{C}(\mathcal{L}(\mathsf{At}))$ that contain only normal probabilistic conditionals.

**Proposition 1.** *If $(\psi \mid \phi)[d]$ is normal then $(\psi \mid \phi)[x]$ is normal for every $x \in [0, 1]$.*

The proof of the above proposition is easy to see as the definition of normality does not depend on the probability of a conditional.

Knowledge bases $\mathcal{K}_1, \mathcal{K}_2$ are *extensionally equivalent*, denoted by $\mathcal{K}_1 \equiv^e \mathcal{K}_2$, if and only if $\mathsf{Mod}(\mathcal{K}_1) = \mathsf{Mod}(\mathcal{K}_2)$. Note that the notion of extensional equivalence does not distinguish between inconsistent knowledge bases, i. e. for inconsistent $\mathcal{K}_1$ and $\mathcal{K}_2$ it always holds that $\mathcal{K}_1 \equiv^e \mathcal{K}_2$. Consequently, we also consider another equivalence relation for knowledge bases. Knowledge bases $\mathcal{K}_1, \mathcal{K}_2$ are *semi-extensionally equivalent*, denoted by $\mathcal{K}_1 \equiv^s \mathcal{K}_2$, if and only if there is a bijection $\rho_{\mathcal{K}_1,\mathcal{K}_2} : \mathcal{K}_1 \to \mathcal{K}_2$ such that $c \equiv^e \rho_{\mathcal{K}_1,\mathcal{K}_2}(c)$ for every $c \in \mathcal{K}_1$. Note that $\mathcal{K}_1 \equiv^s \mathcal{K}_2$ implies $\mathcal{K}_1 \equiv^e \mathcal{K}_2$ but the other direction is not true in general.

*Example 3.* Consider the two knowledge bases $\mathcal{K}_1 = \langle (a)[0.7], (a)[0.4] \rangle$ and $\mathcal{K}_2 = \langle (b)[0.8], (b)[0.3] \rangle$. Both $\mathcal{K}_1$ and $\mathcal{K}_2$ are inconsistent and therefore $\mathcal{K}_1 \equiv^e \mathcal{K}_2$. But it holds that $\mathcal{K}_1 \not\equiv^s \mathcal{K}_2$ as both $(a)[0.7] \not\equiv^e (b)[0.8]$ and $(a)[0.7] \not\equiv^e (b)[0.3]$.

One way for reasoning with knowledge bases is by using model-based inductive reasoning techniques [6]. For example, reasoning based on the *principle of maximum entropy* selects among the models of a knowledge base $\mathcal{K}$ the one unique probability function with maximum entropy. Reasoning with this model satisfies several commonsense properties, see e. g. [6, 5]. However, a necessary requirement for the application of model-based inductive reasoning techniques is the existence of at least one model of a knowledge base. In order to reason with inconsistent knowledge bases the inconsistency has to be resolved first. In the following, we discuss the topic of *inconsistency measurement* for probabilistic conditional logic as inconsistency measures can support the knowledge engineer in the task of resolving inconsistency.

[1] I thank an anonymous reviewer for pointing this formalization out to me.

## 3 Principles for Inconsistency Measurement

An *inconsistency measure* $\mathcal{I}$ is a function that maps a (possibly inconsistent) knowledge base onto a positive real value, i. e. a function $\mathcal{I} : \mathbb{K} \to [0, \infty)$. The value $\mathcal{I}(\mathcal{K})$ for a knowledge base $\mathcal{K}$ is called the *inconsistency value* for $\mathcal{K}$ with respect to $\mathcal{I}$. Intuitively, we want $\mathcal{I}$ to be a function on knowledge bases that is monotonically increasing with the inconsistency in the knowledge base. If the knowledge base is consistent, $\mathcal{I}$ shall be minimal. In order to formalize this intuition we give a list of principles that should be satisfied by any reasonable inconsistency measure. For that we need some further notation.

**Definition 3.** *A set $\mathcal{M}$ is* minimal inconsistent *if $\mathcal{M}$ is inconsistent and every $\mathcal{M}' \subsetneq \mathcal{M}$ is consistent.*

Let $\mathsf{MI}(\mathcal{K})$ be the set of the minimal inconsistent subsets of $\mathcal{K}$.

*Example 4.* Consider the knowledge base $\mathcal{K} = \langle (a)[0.3], (b)[0.5], (a \wedge b)[0.7] \rangle$. Then the set of minimal inconsistent subsets of $\mathcal{K}$ is given via

$$\mathsf{MI}(\mathcal{K}) = \{\ \{(a)[0.3], (a \wedge b)[0.7]\},\ \{(b)[0.5], (a \wedge b)[0.7]\}\ \} \quad .$$

The notion of minimal inconsistent subsets captures those conditionals that are responsible for creating inconsistencies. Conditionals that do not take part in creating an inconsistency are *free*.

**Definition 4.** *A probabilistic conditional $c \in \mathcal{K}$ is* free *in $\mathcal{K}$ if and only if $c \notin \mathcal{M}$ for all $\mathcal{M} \in \mathsf{MI}(\mathcal{K})$.*

For a conditional or a knowledge base $C$ let $\mathsf{At}(C)$ denote the set of atoms appearing in $C$.

**Definition 5.** *A probabilistic conditional $c \in \mathcal{K}$ is* safe *in $\mathcal{K}$ if and only if $\mathsf{At}(c) \cap \mathsf{At}(\mathcal{K} \setminus c) = \emptyset$.*

Note that the notion of safeness is due to Hunter and Konieczny [4]. The notion of a free conditional is clearly more general than the notion of a safe conditional.

**Proposition 2.** *If $c$ is safe in $\mathcal{K}$ then $c$ is free in $\mathcal{K}$.*

The proof of Proposition 2 can be found in [11].

**Definition 6.** *Let $\mathcal{K} \in \mathbb{K}$ be a knowledge base with $\mathcal{K} = \langle c_1, \ldots, c_n \rangle$ and $c_i = (\psi_i \,|\, \phi_i)[d_i]$ for $i = 1, \ldots, n$. The function $\Lambda_{\mathcal{K}} : [0,1]^n \to \mathbb{K}$ with $\Lambda_{\mathcal{K}}(x_1, \ldots, x_n) = \langle (\psi_1 \,|\, \phi_1)[x_1], \ldots, (\psi_n \,|\, \phi_n)[x_n] \rangle$ is called the* characteristic function *of $\mathcal{K}$.*

Due to Proposition 1 the function $\Lambda_{\mathcal{K}}$ is well-defined. The above definition is also the justification for imposing an order on the probabilistic conditionals of a knowledge base.

**Definition 7.** *Let $\mathcal{I}$ be an inconsistency measure and let $\mathcal{K}$ be a knowledge base. The function $\theta_{\mathcal{I},\mathcal{K}} : [0,1]^{|\mathcal{K}|} \to [0,\infty)$ with $\theta_{\mathcal{I},\mathcal{K}} = \mathcal{I} \circ \Lambda_{\mathcal{K}}$ is called the* characteristic inconsistency function *of $\mathcal{I}$ and $\mathcal{K}$.*

Consider now the following properties from [10]. Let $\mathcal{K}, \mathcal{K}'$ be knowledge bases and $c$ a probabilistic conditional.

*Consistency.* $\mathcal{K}$ is consistent if and only if $\mathcal{I}(\mathcal{K}) = 0$
*Monotonicity.* $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K} \cup \{c\})$
*Super-additivity.* If $\mathcal{K} \cap \mathcal{K}' = \emptyset$ then $\mathcal{I}(\mathcal{K} \cup \mathcal{K}') \geq \mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}')$
*Weak independence.* If $c \in \mathcal{K}$ is safe in $\mathcal{K}$ then $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \setminus \{c\})$
*Independence.* If $c \in \mathcal{K}$ is free in $\mathcal{K}$ then $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \setminus \{c\})$
*Penalty.* If $c \in \mathcal{K}$ is not free in $\mathcal{K}$ then $\mathcal{I}(\mathcal{K}) > \mathcal{I}(\mathcal{K} \setminus \{c\})$
*Continuity.* $\theta_{\mathcal{I},\mathcal{K}}$ is continuous

The property *consistency* demands that $\mathcal{I}(\mathcal{K})$ is minimal for consistent $\mathcal{K}$. The properties *monotonicity* and *super-additivity* demand that $\mathcal{I}$ is non-decreasing under the addition of new information. The properties *weak independence* and *independence* say that the inconsistency value should stay the same when adding "harmless" information. The property *penalty* is the counterpart of *independence* and demands that adding inconsistent information increases the inconsistency value. The final property *continuity* describes our main demand for continuous inconsistency measurement, i. e., a "slight" change in the knowledge base should not result in a "vast" change of the inconsistency value.

We also consider the following novel properties. If $f$ is a function $f : [0,1]^n \to [0,\infty)$ then $\nabla f : K \to \mathbb{R}^n$ with $\nabla f(x_1, \ldots, x_n) = (\partial f/\partial x_1, \ldots, \partial f/\partial x_n)$ is its gradient with partial derivatives $\partial f/\partial x_1, \ldots, \partial f/\partial x_n$. There, $K \subseteq [0,1]^n$ is the subset of the domain of $f$ where $f$ is differentiable with respect to all directions.

*Irrelevance of syntax.* If $\mathcal{K}_1 \equiv^s \mathcal{K}_2$ then $\mathcal{I}(\mathcal{K}_1) = \mathcal{I}(\mathcal{K}_2)$
*MI-separability.* If $\mathsf{MI}(\mathcal{K}_1 \cup \mathcal{K}_2) = \mathsf{MI}(\mathcal{K}_1) \cup \mathsf{MI}(\mathcal{K}_2)$ and $\mathsf{MI}(\mathcal{K}_1) \cap \mathsf{MI}(\mathcal{K}_2) = \emptyset$ then $\mathcal{I}(\mathcal{K}_1 \cup \mathcal{K}_2) = \mathcal{I}(\mathcal{K}_1) + \mathcal{I}(\mathcal{K}_2)$
*Differentiability.* $\theta_{\mathcal{I},\mathcal{K}}$ is differentiable in $(0,1)^{|\mathcal{K}|}$
*Weak differentiability.* $\theta_{\mathcal{I},\mathcal{K}}$ is differentiable almost everywhere in $(0,1)^{|\mathcal{K}|}$
*Sub-linearity.* $\mathsf{Im}\, \nabla\theta_{\mathcal{I},\mathcal{K}} \subseteq [-1,1]^{|\mathcal{K}|}$

We define the property *irrelevance of syntax* in terms of the equivalence relation $\equiv^s$ as all inconsistent knowledge bases are equivalent with respect to $\equiv^e$. For an inconsistency measure $\mathcal{I}$, imposing *irrelevance of syntax* to hold in terms of $\equiv^e$ would yield $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K}')$ for every two inconsistent knowledge bases $\mathcal{K}, \mathcal{K}'$. The property *MI-separability*—which has been adapted from [3]—states that determining the value of $\mathcal{I}(\mathcal{K}_1 \cup \mathcal{K}_2)$ can be split into determining the values of $\mathcal{I}(\mathcal{K}_1)$ and $\mathcal{I}(\mathcal{K}_2)$ if the minimal inconsistent subsets of $\mathcal{K}_1 \cup \mathcal{K}_2$ are partitioned by $\mathcal{K}_1$ and $\mathcal{K}_2$. The property *differentiability* strengthens the property *continuity* and expects $\mathcal{I}$ to behave even more smoothly. The property *weak differentiability* allows $\mathcal{I}$ to be non-differentiable on a null set. Finally, the property *sub-linearity* demands that the value $\mathcal{I}(\mathcal{K})$ changes at most linearly in the change of $\mathcal{K}$. This

means, for example, that if one changes the probability of a conditional in $\mathcal{K}$ by some value $\alpha$, then the difference between the corresponding values of $\mathcal{I}$ should not be more than $\alpha$.

Some relationships between the above properties are as follows.

**Proposition 3.** *Let $\mathcal{I}$ be an inconsistency measure and let $\mathcal{K}, \mathcal{K}'$ be some knowledge bases.*

1. *If $\mathcal{I}$ satisfies* super-additivity *then $\mathcal{I}$ satisfies* monotonicity*.*
2. *If $\mathcal{I}$ satisfies* independence *then $\mathcal{I}$ satisfies* weak independence*.*
3. *If $\mathcal{I}$ satisfies* MI-separability *then $\mathcal{I}$ satisfies* independence*.*
4. *If $\mathcal{I}$ satisfies* differentiability *then $\mathcal{I}$ satisfies* continuity*.*
5. *If $\mathcal{I}$ satisfies* differentiability *then $\mathcal{I}$ satisfies* weak differentiability*.*
6. *$\mathcal{K} \subseteq \mathcal{K}'$ implies $\mathsf{MI}(\mathcal{K}) \subseteq \mathsf{MI}(\mathcal{K}')$.*
7. *If $\mathcal{I}$ satisfies* independence *then $\mathsf{MI}(\mathcal{K}) = \mathsf{MI}(\mathcal{K}')$ implies $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K}')$.*
8. *If $\mathcal{I}$ satisfies* independence *and* penalty *then $\mathsf{MI}(\mathcal{K}) \subsetneq \mathsf{MI}(\mathcal{K}')$ implies $\mathcal{I}(\mathcal{K}) < \mathcal{I}(\mathcal{K}')$.*

The proofs of 1.)-3.) and 6.)-8.) can be found in [11]. The proofs of 4.) and 5.) are obvious.

Previous research on inconsistency measurement focuses on inconsistency measurement on propositional logic, see e. g. [3]. Adopting those measures for probabilistic conditional logic is straightforward [11]. For example, consider the following definition.

**Definition 8.** *The function $\mathcal{I}^{\#} : \mathbb{K} \to [0, \infty)$ defined via $\mathcal{I}^{\#}(\mathcal{K}) = |\mathsf{MI}(\mathcal{K})|$ is called the* MI cardinality measure*.*

The MI cardinality measure determines the inconsistency value of a knowledge base $\mathcal{K}$ as the number of minimal inconsistent subsets of $\mathcal{K}$.

*Example 5.* We continue Ex. 4. There it holds that $\mathcal{I}^{\#}(\mathcal{K}) = 2$.

Although $\mathcal{I}^{\#}$ is a rather simple inconsistency measure it already complies with many principles.

**Proposition 4.** *The function $\mathcal{I}^{\#}$ satisfies* consistency*,* monotonicity*,* super-additivity*,* weak independence*,* independence*,* MI-separability*, and* penalty*.*

The proof of Proposition 4 can be found in [11]. However, as the following example shows, the MI cardinality measure—and other inconsistency measures that were developed for propositional logic—does not satisfy *continuity* which is a major drawback for the probabilistic setting.

*Example 6.* Consider the knowledge base $\mathcal{K} = \langle (b \mid a)[1], (a)[1], (b)[0] \rangle$ which models strongly inconsistent information. Clearly, it holds that $\mathcal{I}^{\#}(\mathcal{K}) = 1$. Consider now the two modifications $\mathcal{K}', \mathcal{K}''$ of $\mathcal{K}$ given via

$$\mathcal{K}' = \langle (b \mid a)[0.6], (a)[0.6], (b)[0.3599] \rangle$$
$$\mathcal{K}'' = \langle (b \mid a)[0.6], (a)[0.6], (b)[0.36] \rangle \quad .$$

It is also clear that $\mathcal{I}^{\#}(\mathcal{K}') = 1$ and $\mathcal{I}^{\#}(\mathcal{K}'') = 0$. By comparing $\mathcal{K}'$ and $\mathcal{K}''$ one can discover only a minor difference of the modeled knowledge. From a practical point of view, whether $b$ has probability 0.3599 or 0.36 may not matter for the intended application. Still, a knowledge engineer may not grasp the harmlessness of the inconsistency in $\mathcal{K}'$ as $\mathcal{K}'$ and $\mathcal{K}$ have the same inconsistency value.

In the following, we discuss inconsistency measures that are more apt for the probabilistic setting.

## 4 Measuring Inconsistency by Distance Minimization

As can be seen in Ex. 6 the probabilities of conditionals play a crucial role in creating inconsistencies. In order to respect this role we propose a family of inconsistency measures that is based on the distance to consistency. Afterwards we propose a compound measure that uses this measure and behaves well with the desired properties.

Before defining the measure we need some further notation. Knowledge bases $\mathcal{K}_1, \mathcal{K}_2$ are *qualitatively equivalent*, denoted by $\mathcal{K}_1 \cong^q \mathcal{K}_2$, if and only if there is a bijection $\sigma_{\mathcal{K}_1,\mathcal{K}_2} : \mathcal{K}_1 \to \mathcal{K}_2$ such that $\mathsf{body}(c) \equiv \mathsf{body}(\sigma(c))$ and $\mathsf{head}(c) \wedge \mathsf{body}(c) \equiv \mathsf{head}(\sigma(c)) \wedge \mathsf{body}(\sigma(c))$ for every $c \in \mathcal{K}_1$. Note that the function $\sigma_{\mathcal{K}_1,\mathcal{K}_2}$ might not be uniquely determined.

*Example 7.* Consider the knowledge bases $\mathcal{K}_1 = \langle (a)[0.2], (a)[0.8] \rangle$ and $\mathcal{K}_2 = \langle (a)[0.3], (a)[0.9] \rangle$. It holds that $\mathcal{K}_1 \cong^q \mathcal{K}_2$ but there are two bijections $\sigma^1_{\mathcal{K}_1,\mathcal{K}_2}$ and $\sigma^2_{\mathcal{K}_1,\mathcal{K}_2}$ given via

$$\sigma^1_{\mathcal{K}_1,\mathcal{K}_2}((a)[0.2]) = (a)[0.3] \qquad \sigma^1_{\mathcal{K}_1,\mathcal{K}_2}((a)[0.8]) = (a)[0.9]$$
$$\sigma^2_{\mathcal{K}_1,\mathcal{K}_2}((a)[0.2]) = (a)[0.9] \qquad \sigma^1_{\mathcal{K}_1,\mathcal{K}_2}((a)[0.8]) = (a)[0.3]$$

that establish the qualitative equivalence of $\mathcal{K}_1$ and $\mathcal{K}_2$.

If $\mathcal{K}_1 \cong^q \mathcal{K}_2$ let $\mathcal{S}_{\mathcal{K}_1,\mathcal{K}_2}$ be the set of bijections between $\mathcal{K}_1$ and $\mathcal{K}_2$ with the above property. Note that $\mathcal{S}_{\mathcal{K}_1,\mathcal{K}_2}$ is finite as both $\mathcal{K}_1$ and $\mathcal{K}_2$ are finite. Let $\mathbb{N}^+$ denote the set of positive integers.

**Definition 9.** *Let $\mathcal{K}_1, \mathcal{K}_2$ be some knowledge bases and let $p \in \mathbb{N}^+$. Then the* $p$-norm distance *$d^p(\mathcal{K}_1, \mathcal{K}_2)$ of $\mathcal{K}_1$ to $\mathcal{K}_2$ is defined via*

$$d^p(\mathcal{K}_1, \mathcal{K}_2) = \begin{cases} \min\limits_{\sigma \in \mathcal{S}_{\mathcal{K}_1,\mathcal{K}_2}} \left\{ \sqrt[p]{\sum_{c \in \mathcal{K}_1} |\mathsf{prob}(c) - \mathsf{prob}(\sigma(c))|^p} \right\} & \text{if } \mathcal{K}_1 \cong^q \mathcal{K}_2 \\ \infty & \text{otherwise} \end{cases}.$$

Note that $d^p$ is indeed a distance measure, i. e., it is positive definite, symmetric, and satisfies the triangle inequality. This measure assigns an infinite distance to two knowledge bases $\mathcal{K}_1, \mathcal{K}_2$ iff $\mathcal{K}_1, \mathcal{K}_2$ are not qualitatively equivalent. Otherwise it is equivalent to the standard $p$-norm distance by interpreting probabilities of conditionals as coordinates and selecting a bijection $\sigma \in \mathcal{S}_{\mathcal{K}_1,\mathcal{K}_2}$ that minimizes this distance.

*Example 8.* For $\mathcal{K}_1$ and $\mathcal{K}_2$ as given in Ex. 7 it holds that $d^1(\mathcal{K}_1, \mathcal{K}_2) = 0.2$ and $d^2(\mathcal{K}_1, \mathcal{K}_2) \approx 0.1414$. Note that $\sigma^1_{\mathcal{K}_1,\mathcal{K}_2}$ is used for determining $d^p(\mathcal{K}_1, \mathcal{K}_2)$ as $\sigma^2_{\mathcal{K}_1,\mathcal{K}_2}$ yields values 1.2 and $\approx 0.8602$, respectively.

The following definition has been rephrased from [10, 7].

**Definition 10.** *Let $\mathcal{K}$ be a knowledge base and let $p \in \mathbb{N}^+$. Then define the* $d^p$-measure $\mathcal{I}^p$ *via*

$$\mathcal{I}^p(\mathcal{K}) = \min\{d^p(\mathcal{K}, \mathcal{K}') \mid \mathcal{K}' \text{ consistent}\} \tag{1}$$

*for a knowledge base $\mathcal{K}$.*

The value $\mathcal{I}^p(\mathcal{K})$ is the minimal distance to a knowledge base $\mathcal{K}'$ that is both qualitatively equivalent to $\mathcal{K}$ and consistent. Now we can also justify representing knowledge bases as multi-sets. Considering the knowledge base $\mathcal{K} = \langle (a)[0.2], (a)[0.6] \rangle$, it holds that $\mathcal{K}' = \langle (a)[0.4], (a)[0.4] \rangle$ minimizes the $p$-norm distance to $\mathcal{K}$.

The above definition presupposes that the minimum in Equation (1) exists. The following proposition shows that this is indeed the case.

**Proposition 5.** *The function $\mathcal{I}^p$ is well-defined.*

*Proof. Let $\mathcal{K} = \langle (\psi_1 \mid \phi_1)[d_1], \ldots, (\psi_n \mid \phi_n)[d_n] \rangle$ be a knowledge base and let $P_0$ be the uniform probability function on $\Omega(\mathsf{At})$, i. e, it holds that $P_0(\omega) = 1/|\Omega(\mathsf{At})|$ for every $\omega \in \Omega(\mathsf{At})$ (note that $\Omega(\mathsf{At})$ is finite as $\mathsf{At}$ is finite). Let $\mathcal{K}'$ be the knowledge base defined via*

$$\mathcal{K}' = \langle (\psi_1 \mid \phi_1)[P_0(\psi_1 \mid \phi_1)], \ldots, (\psi_n \mid \phi_n)[P_0(\psi_n \mid \phi_n)] \rangle$$

*As $P_0$ is a positive probability function and every $c \in \mathcal{K}$ is normal it follows that $\mathcal{K}'$ is well-defined and $P_0 \models^{pr} \mathcal{K}'$. As $\mathcal{K} \cong^q \mathcal{K}'$ it follows that $\mathcal{I}^p(\mathcal{K})$ is finite. Furthermore, observe that the set*

$$D_{\mathcal{K}} = \{\langle x_1, \ldots, x_n \rangle \mid \langle (\psi_1 \mid \phi_1)[x_1], \ldots, (\psi_n \mid \phi_n)[x_n] \rangle \text{ is consistent } \}$$

*is compact (bounded and closed) as probabilistic satisfaction is defined via the equation $P(\psi\phi) = dP(\phi)$ (for a probabilistic conditional $(\psi \mid \phi)[d]$). As the functional mapping*

$$\langle x_1, \ldots, x_n \rangle \mapsto \sqrt[p]{|d_1 - x_1|^p + \ldots + |d_n - x_n|^p}$$

*is continuous it follows that the set $\{d^p(\mathcal{K}, \mathcal{K}') \mid \mathcal{K}' \text{ consistent}\}$ is closed. Therefore, the minimum of this set and the value of $\mathcal{I}^p$ is well-defined.* □

In [7] it has been shown that for every $p, p' \in \mathbb{N}^+$ with $p \neq p'$ the two measures $\mathcal{I}^p$ and $\mathcal{I}^{p'}$ are not equivalent, i. e., there are knowledge bases $\mathcal{K}_1$ and $\mathcal{K}_2$ such that $\mathcal{I}^p(\mathcal{K}_1) > \mathcal{I}^p(\mathcal{K}_2)$ but $\mathcal{I}^{p'}(\mathcal{K}_1) < \mathcal{I}^{p'}(\mathcal{K}_2)$.

*Example 9.* We continue Ex. 6. There, the knowledge base $\mathcal{K}^* = \langle (b \,|\, a)[1], (a)[0.5], (b)[0.5] \rangle$ satisfies $\mathcal{I}^p(\mathcal{K}) = d^p(\mathcal{K}, \mathcal{K}^*)$ for every $p$. In particular, it holds that $\mathcal{I}^p(\mathcal{K}) = \sqrt[p]{2 \cdot 0.5^p}$. For example, it holds that $\mathcal{I}^1(\mathcal{K}) = 1$ and $\mathcal{I}^2(\mathcal{K}) \approx 0.707$. Furthermore, it holds that $\mathcal{I}^1(\mathcal{K}') = 0.0001$ and $\mathcal{I}^2(\mathcal{K}') \approx 0.00006$, and clearly $\mathcal{I}^1(\mathcal{K}'') = \mathcal{I}^2(\mathcal{K}'') = 0$.

We also propose the following compound measure that explicitly considers the crucial role of minimal inconsistent subsets.

**Definition 11.** *Let $\mathcal{K}$ be a knowledge base and let $\mathcal{I}$ be an inconsistency measure. Then define the* MI-measure $\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}(\mathcal{K})$ *of $\mathcal{K}$ and $\mathcal{I}$ via*

$$\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}(\mathcal{K}) = \sum_{\mathcal{M} \in \mathsf{MI}(\mathcal{K})} \mathcal{I}(\mathcal{M}) \quad .$$

The MI-measure is defined as the sum of the inconsistency values of all minimal inconsistent subsets of the knowledge base under consideration. In the next section, we investigate the properties of the measures proposed above.

## 5 Analysis and Comparison

We first investigate the properties of the $d^p$-measure. We can extend a result from [10] as follows.

**Theorem 1.** *If $p \in \mathbb{N}^+$ then $\mathcal{I}^p$ satisfies* consistency, monotonicity, weak independence, independence, irrelevance of syntax, continuity, weak differentiability, *and* sub-linearity.

*Proof.*

Consistency. *As $\mathcal{K}$ is consistent and $d^p(\mathcal{K}, \mathcal{K}) = 0$ it follows directly $\mathcal{I}^p(\mathcal{K}) = 0$.*

Monotonicity. *Let $\mathcal{K} = \langle c_1, \ldots, c_n \rangle$ and let $\mathcal{K}'$ be consistent and $d^p(\mathcal{K}, \mathcal{K}') = \mathcal{I}^p(\mathcal{K})$. Let furthermore $\sigma_{\mathcal{K},\mathcal{K}'} \in \mathcal{S}_{\mathcal{K}_1,\mathcal{K}_2}$ be the bijection used to determine $d^p(\mathcal{K}, \mathcal{K}')$. It follows that $\mathcal{K}'' = \mathcal{K}' \setminus \{\sigma_{\mathcal{K},\mathcal{K}'}(c_n)\}$ is consistent as well and $\mathcal{K} \setminus \{c_n\} \cong^q \mathcal{K}''$. It follows that $\mathcal{I}^p(\mathcal{K} \setminus \{c_n\}) \leq d^p(\mathcal{K} \setminus \{c_n\}, \mathcal{K}'')$. Setting $a_i = |\mathsf{prob}(c_i) - \mathsf{prob}(\sigma_{\mathcal{K},\mathcal{K}'}(c_i))|$ for $i = 1, \ldots, n$ we get*

$$\begin{aligned} \mathcal{I}^p(\mathcal{K}) = d^p(\mathcal{K}, \mathcal{K}') &= \sqrt[p]{a_1^p + \ldots + a_n^p} \\ &\geq \sqrt[p]{a_1^p + \ldots + a_{n-1}^p} = d^p(\mathcal{K} \setminus \{c\}, \mathcal{K}'') \geq \mathcal{I}^p(\mathcal{K} \setminus \{c_n\}) \end{aligned}$$

Independence. *In [11] it has been shown that $\mathcal{I}^p$ for $p = 1$ satisfies* independence. *This result can be extended to arbitrary $p$ in a straightforward fashion.*

Irrelevance of syntax. *Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be knowledge bases with $\mathcal{K}_1 \equiv^s \mathcal{K}_2$. Let $\mathcal{K}_1'$ be consistent such that $\mathcal{I}^p(\mathcal{K}_1) = d^p(\mathcal{K}_1, \mathcal{K}_1')$. It follows that $\mathcal{K}_1 \cong^q \mathcal{K}_1'$. As $\mathcal{K}_1 \equiv^s \mathcal{K}_2$ there is a consistent $\mathcal{K}_2'$ such that $\mathcal{K}_1' \equiv^s \mathcal{K}_2'$ and $\mathcal{K}_2 \cong^q \mathcal{K}_2'$. It follows that $\mathcal{I}^p(\mathcal{K}_2) \leq d^p(\mathcal{K}_2, \mathcal{K}_2') = d^p(\mathcal{K}_1, \mathcal{K}_1') = \mathcal{I}^p(\mathcal{K}_1)$. Similarly we obtain $\mathcal{I}^p(\mathcal{K}_1) \leq \mathcal{I}^p(\mathcal{K}_2)$ and therefore the claim.*

Continuity. *In [11] it has been shown that $\mathcal{I}^p$ for $p = 1$ satisfies* continuity. *This result can be extended to arbitrary $p$ in a straightforward fashion.*

Weak differentiability. *We only give a proof sketch for* weak differentiability. *Let $\vec{x} \in (0,1)^{|\mathcal{K}|}$ such that there is an open $\epsilon$-ball $B_\epsilon$ with $\vec{x} \in B_\epsilon$ and $\theta_{\mathcal{I}^p,\mathcal{K}}(\vec{y}) > 0$ for every $\vec{y} \in B_\epsilon$. Then $\theta_{\mathcal{I}^p,\mathcal{K}}$ is differentiable on $B_\epsilon$ as the $p$-norm distance is a differentiable function if the distance does not equal zero. Furthermore, let now $\vec{x} \in (0,1)^{|\mathcal{K}|}$ be such that there is an open $\epsilon$-ball $B_\epsilon$ with $\vec{x} \in B_\epsilon$ and $\theta_{\mathcal{I}^p,\mathcal{K}}(\vec{y}) = 0$ for every $\vec{y} \in B_\epsilon$. Then $\theta_{\mathcal{I}^p,\mathcal{K}}$ is differentiable on $B_\epsilon$ as it is a constant function. Note furthermore that the set $C \subseteq (0,1)^{|\mathcal{K}|}$ such that for every $\vec{y} \in C$ it holds that $\theta_{\mathcal{I}^p,\mathcal{K}}(\vec{y}) = 0$ is the finite union of pair-wise disjoint closed convex sets $F_1, \ldots, F_m$, cf. [11]. Without loss of generality, let $F_1, \ldots, F_k$ with $k \leq m$ be the sets with dimension $|\mathcal{K}|$ and $F_{k+1}, \ldots, F_m$ be the sets with a dimension less than $|\mathcal{K}|$. Let* bd *$S$ denote the boundary of a set $S$. Note that* bd *$F_i$ has dimension $|\mathcal{K}| - 1$ for $i = 1, \ldots, k$. Then the set $F =$* bd *$F_1 \cup \ldots$* bd *$F_k \cup F_{k+1} \cup \ldots F_m$ is a null set in $(0,1)^{|\mathcal{K}|}$ and $\theta_{\mathcal{I}^p,\mathcal{K}}$ is differentiable on $(0,1)^{|\mathcal{K}|} \setminus F$.*

Sub-linearity. *We only give a proof sketch for* sub-linearity. *Let $\mathcal{K}$ be the knowledge base $\mathcal{K} = \langle (\psi_1 \,|\, \phi_1)[d_1], \ldots, (\psi_n \,|\, \phi_n)[d_n] \rangle$ and let $\vec{x} \in [0,1]^{|\mathcal{K}|}$ such that $\theta_{\mathcal{I},\mathcal{K}}$ is differentiable in $\vec{x} = (x_1, \ldots, x_n)$. Note that*

$$\left( \sqrt[p]{g(x)} \right)' = \frac{1}{p} \frac{1}{g(x)^{p-1}} g'(x)$$

*for differentiable $g$ and that $|(|x|)'| = 1$ for $x \neq 0$. Then consider the function*

$$f(\vec{x}) = \sqrt[p]{|d_1 - x_1|^p + \ldots + |d_n - x_n|^p} \tag{2}$$

*and the following bound on the absolute value of its partial derivatives ($i = 1, \ldots, n$)*

$$\begin{aligned}
\left| \frac{\partial f}{\partial x_i} \right| &= \left| \frac{1}{p} \frac{1}{\left( \sqrt[p]{|d_1 - x_1|^p + \ldots + |d_n - x_n|^p} \right)^{p-1}} \cdot p \cdot |d_i - x_i|^{p-1} \right| \\
&= \left| \left( \frac{|d_i - x_i|}{\sqrt[p]{|d_1 - x_1|^p + \ldots + |d_n - x_n|^p}} \right)^{p-1} \right| \\
&= \left| \left( \sqrt[p]{\frac{|d_i - x_i|^p}{|d_1 - x_1|^p + \ldots + |d_n - x_n|^p}} \right)^{p-1} \right| \\
&\leq 1
\end{aligned}$$

*The above means that $d^p(\mathcal{K}, \mathcal{K}')$ is sub-linear in $\mathcal{K}'$ for fixed $\mathcal{K}$. Assume now that there is an open $\epsilon$-ball $B_\epsilon$ with $\vec{x} \in B_\epsilon$ and $\theta_{\mathcal{I}^p,\mathcal{K}}(\vec{y}) > 0$ for every $\vec{y} \in B_\epsilon$. Then $|\partial \theta_{\mathcal{I}^p,\mathcal{K}} / \partial x_i| \leq 1$ directly from above (as $\theta_{\mathcal{I}^p,\mathcal{K}}$ behaves like $f$ in the worst case). Furthermore, let now $\vec{x} \in (0,1)^{|\mathcal{K}|}$ be such that there is an open $\epsilon$-ball $B_\epsilon$ with $\vec{x} \in B_\epsilon$ and $\theta_{\mathcal{I}^p,\mathcal{K}}(\vec{y}) = 0$ for every $\vec{y} \in B_\epsilon$. Then clearly $|\partial \theta_{\mathcal{I}^p,\mathcal{K}} / \partial x_i| = 0 \leq 1$.*

*Due to Proposition 3 it also follows that $\mathcal{I}^p$ satisfies* weak independence. □

Note that $\mathcal{I}^p$ does not satisfy *differentiability* in general as the following example shows.

*Example 10.* Consider the knowledge base $\mathcal{K} = \langle (a)[0.7], (a)[0.3] \rangle$. It is easy to see that e. g. $\theta_{\mathcal{I}^1,\mathcal{K}}(x, y) = |x - y|$. In particular, it holds that $\theta_{\mathcal{I}^1,\mathcal{K}}(x, y) = 0$ if and only if $x = y$. It also also quite clear that the absolute value $|x|$ is continuous for all $x$ but only differentiable for $x \neq 0$.

However, for $p > 1$ we can strengthen Theorem 1 as follows.

**Theorem 2.** *If $p \in \mathbb{N}^+$ and $p > 1$ then $\mathcal{I}^p$ satisfies* differentiability.

We omit the proof of the above theorem due to space restrictions but note that this follows from the differentiability of the $p$-norm distance for $p > 1$. Observe that $\mathcal{I}^p$ does not satisfy *penalty* which has been mistakenly claimed in [10]. Consider the following counterexample.

*Example 11.* Consider the knowledge base $\mathcal{K} = \langle (a)[0.7], (a)[0.3] \rangle$ and the probabilistic conditional $(a)[0.5]$. Then $(a)[0.5]$ is not free in $\mathcal{K}' = \mathcal{K} \cup \{(a)[0.5]\}$ as $\{(a)[0.3], (a)[0.5]\} \in \mathsf{MI}(\mathcal{K}')$. However, it holds that $\mathcal{I}^1(\mathcal{K}) = \mathcal{I}^1(\mathcal{K}') = 0.4$—as $\langle (a)[0.5], (a)[0.5] \rangle$ has minimal distance to $\mathcal{K}$ and $\langle (a)[0.5], (a)[0.5], (a)[0.5] \rangle$ has minimal distance to $\mathcal{K}'$—which violates *penalty*.

In [11] it has been show that $\mathcal{I}^p$ for $p = 1$ also satisfies *MI-separability* and *super-additivity*. This is not true for arbitrary values of $p$ as the following example shows.

*Example 12.* Let $\mathcal{K} = \langle (a)[0.3], (a)[0.7], (b)[0.3], (b)[0.7] \rangle$. It is easy to see that $\mathcal{I}^2(\mathcal{K}) = \sqrt{0.2^2 + 0.2^2 + 0.2^2 + 0.2^2} = 0.4$. It also holds that

$$\mathcal{I}^2(\langle (a)[0.3], (a)[0.7] \rangle) = \mathcal{I}^2(\langle (b)[0.3], (b)[0.7] \rangle) = \sqrt{0.2^2 + 0.2^2} \approx 0.283 \quad .$$

It follows that

$$\mathcal{I}^2(\mathcal{K}) < \mathcal{I}^2(\langle (a)[0.3], (a)[0.7] \rangle) + \mathcal{I}^2(\langle (b)[0.3], (b)[0.7] \rangle)$$

violating *super-additivity* and *MI-separability* as $\langle (a)[0.3], (a)[0.7] \rangle$ and $\langle (b)[0.3], (b)[0.7] \rangle$ partition the set of minimal inconsistent subsets of $\mathcal{K}$.

We now have a look at the properties of the MI-measure.

**Theorem 3.** *Let $\mathcal{I}$ be an inconsistency measure.*

1. *$\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}$ satisfies* monotonicity, super-additivity, weak independence, independence, *and* MI-separability.
2. *If $\mathcal{I}$ satisfies* consistency *then $\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}$ satisfies* consistency *and* penalty.
3. *If $\mathcal{I}$ satisfies* irrelevance of syntax *then $\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}$ satisfies* irrelevance of syntax.
4. *If $\mathcal{I}$ satisfies* continuity *then $\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}$ satisfies* continuity.

5. *If $\mathcal{I}$ satisfies* differentiability *then $\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}$ satisfies* differentiability.
6. *If $\mathcal{I}$ satisfies* weak differentiability *then $\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}$ satisfies* weak differentiability.

*Proof.*

1. *We first show that $\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}$ satisfies* super-additivity. *If $\mathcal{K} \cap \mathcal{K}' = \emptyset$ then it holds that $\mathsf{MI}(\mathcal{K}) \cap \mathsf{MI}(\mathcal{K}') = \emptyset$ as well. Due to 6.) in Proposition 3 it follows that $\mathsf{MI}(\mathcal{K}) \cup \mathsf{MI}(\mathcal{K}') \subseteq \mathsf{MI}(\mathcal{K} \cup \mathcal{K}')$. It follows*

$$\begin{aligned}\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}(\mathcal{K} \cup \mathcal{K}') &= \sum_{\mathcal{M} \in \mathsf{MI}(\mathcal{K} \cup \mathcal{K}')} \mathcal{I}(\mathcal{M}) \geq \sum_{\mathcal{M} \in \mathsf{MI}(\mathcal{K})} \mathcal{I}(\mathcal{M}) + \sum_{\mathcal{M} \in \mathsf{MI}(\mathcal{K}')} \mathcal{I}(\mathcal{M}) \\ &= \mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}(\mathcal{K}) + \mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}(\mathcal{K}') \quad .\end{aligned}$$

   *Due to 1.) in Proposition 3 it also follows that $\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}$ satisfies* monotonicity. *We now show that $\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}$ satisfies* $\mathsf{MI}$-separability. *Let $\mathsf{MI}(\mathcal{K} \cup \mathcal{K}') = \mathsf{MI}(\mathcal{K}) \cup \mathsf{MI}(\mathcal{K}')$ and $\mathsf{MI}(\mathcal{K}) \cap \mathsf{MI}(\mathcal{K}') = \emptyset$. Then clearly*

$$\begin{aligned}\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}(\mathcal{K} \cup \mathcal{K}') &= \sum_{\mathcal{M} \in \mathsf{MI}(\mathcal{K} \cup \mathcal{K}')} \mathcal{I}(\mathcal{M}) = \sum_{\mathcal{M} \in \mathsf{MI}(\mathcal{K})} \mathcal{I}(\mathcal{M}) + \sum_{\mathcal{M} \in \mathsf{MI}(\mathcal{K}')} \mathcal{I}(\mathcal{M}) \\ &= \mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}(\mathcal{K}) + \mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}(\mathcal{K}') \quad .\end{aligned}$$

   *Due to 2.) and 3.) in Proposition 3 it also follows that $\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}$ satisfies* independence *and* weak independence.
2. *We first show that $\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}$ satisfies* consistency. *If $\mathcal{K}$ is consistent then $\mathsf{MI}(\mathcal{K}) = \emptyset$ and $\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}(\mathcal{K}) = 0$. If $\mathcal{K}$ is inconsistent then there is a $\mathcal{M} \in \mathsf{MI}(\mathcal{K})$ and as $\mathcal{I}$ satisfies* consistency *it follows that $\mathcal{I}(\mathcal{M}) > 0$. Hence, $\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}(\mathcal{K}) > 0$ as well. We now show that $\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}$ satisfies* penalty. *Let $c \in \mathcal{K}$ be a probabilistic conditional that is not free in $\mathcal{K}$. Due to 6.) in Proposition 3 it follows that $\mathsf{MI}(\mathcal{K} \setminus \{c\}) \subseteq \mathsf{MI}(\mathcal{K})$. As $c \notin \mathcal{K} \setminus \{c\}$ and there is at least one $\mathcal{M} \in \mathsf{MI}(\mathcal{K})$ with $c \in \mathcal{M}$ it follows that $\mathsf{MI}(\mathcal{K} \setminus \{c\}) \subsetneq \mathsf{MI}(\mathcal{K})$. As $\mathcal{I}$ satisfies* consistency *it follows that $\mathcal{I}(\mathcal{M}) > 0$ and therefore $\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}(\mathcal{K} \setminus \{c\}) < \mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}(\mathcal{K})$.*
3. *Let it hold that $\mathcal{K}_1 \equiv^s \mathcal{K}_2$. It follows that for every $\mathcal{M} \in \mathsf{MI}(\mathcal{K}_1)$ there is $\mathcal{M}' \in \mathsf{MI}(\mathcal{K}_2)$ with $\mathcal{M} \equiv^s \mathcal{M}'$, and vice versa. As $\mathcal{I}$ satisfies* irrelevance of syntax *it follows that $\mathcal{I}(\mathcal{M}) = \mathcal{I}(\mathcal{M}')$ for every $\mathcal{M} \in \mathsf{MI}(\mathcal{K}_1)$. Hence, it holds that $\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}(\mathcal{K}_1) = \sum_{\mathcal{M} \in \mathsf{MI}(\mathcal{K}_1)} \mathcal{I}(\mathcal{M}') = \sum_{\mathcal{M}' \in \mathsf{MI}(\mathcal{K}_2)} \mathcal{I}(\mathcal{M}') = \mathcal{I}^{\mathcal{I}}_{\mathsf{MI}}(\mathcal{K}_2)$.*
4. *It is easy to see that $\theta_{\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}},\mathcal{K}}$ is given via $\theta_{\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}},\mathcal{K}} = \sum_{\mathcal{M} \in \mathsf{MI}(\mathcal{K})} \theta_{\mathcal{I},\mathcal{M}}$ (given an adequate ordering of the conditionals in $\mathcal{K}$). It follows directly, that $\theta_{\mathcal{I}^{\mathcal{I}}_{\mathsf{MI}},\mathcal{K}}$ is continuous if $\theta_{\mathcal{I},\mathcal{M}}$ is continuous for every $\mathcal{M} \in \mathsf{MI}(\mathcal{K})$, i. e., if $\mathcal{I}$ satisfies* continuity.
5. *This holds due to the same argument used in 4.).*
6. *This holds due to the same argument used in 4.).* □

As one can see the $\mathsf{MI}$-measure behaves very well with respect to our rationality postulates and even satisfies *penalty*, provided that the inner measure satisfies *consistency*.

*Example 13.* We continue Ex. 11. There it is $\mathcal{I}^{\mathcal{I}^1}_{\mathsf{MI}}(\mathcal{K}) = \mathcal{I}^p(\mathcal{K}) = 0.4$ but

$$\begin{aligned}\mathcal{I}^{\mathcal{I}^1}_{\mathsf{MI}}(\mathcal{K}') = \mathcal{I}^1(\langle (a)[0.7], (a)[0.3]\rangle) + \mathcal{I}^1(\langle (a)[0.7], (a)[0.5]\rangle) \\ + \mathcal{I}^1(\langle (a)[0.3], (a)[0.5]\rangle) = 0.4 + 0.2 + 0.2 = 0.8 \quad .\end{aligned}$$

Therefore, the addition of the conditional $(a)[0.5]$ is penalized by $\mathcal{I}^{\mathcal{I}^1}_{\mathsf{MI}}$.

The following corollary is a direct application of Theorems 1 and 3.

**Corollary 1.** *If $p \in \mathbb{N}^+$ then $\mathcal{I}^{\mathcal{I}^p}_{\mathsf{MI}}$ satisfies* consistency*,* monotonicity*,* super-additivity*,* weak independence*,* independence*,* MI-separability*,* penalty*,* irrelevance of syntax*,* continuity*, and* weak differentiability*. If $p > 1$ then $\mathcal{I}^{\mathcal{I}^p}_{\mathsf{MI}}$ also satisfies* differentiability*.*

Note that $\mathcal{I}^{\mathcal{I}^p}_{\mathsf{MI}}$ does not satisfy *sub-linearity* in general. Consider the following counterexample.

*Example 14.* Consider the knowledge base $\mathcal{K} = \langle (a)[0.7], (a)[0.3], (\neg a)[0.7]\rangle$. Note that for $x, y, z \in [0, 1]$ there are three (potential) minimal inconsistent subsets of $\Lambda_{\mathcal{K}}(x, y, z)$: $\{(a)[x], (a)[y]\}$, $\{(a)[x], (\neg a)[z]\}$, $\{(a)[y], (\neg a)[z]\}$. Then $\theta_{\mathcal{I}^1,\mathcal{K}}$ amounts to $\theta_{\mathcal{I}^1,\mathcal{K}}(x, y, z) = |x - y| + |1 - x - z| + |1 - y - z|$. For $x = y = 0$ and $z = 1$ we get $\theta_{\mathcal{I}^1,\mathcal{K}}(x, y, z) = 0$ and for $x = y = z = 0$ we get $\theta_{\mathcal{I}^1,\mathcal{K}}(x, y, z) = 2$. It follows that the absolute value of the partial derivation of $\theta_{\mathcal{I}^1,\mathcal{K}}$ with respect to the third coordinate has to be larger than 1 for at least one point.

## 6 Related Work

The work reported in this paper is based on results from [10, 7]. We extended the investigation of measuring inconsistency from [10] by introducing several novel rationality postulates, the MI-measure, and the resulting technical discussion. The $d^p$-measure has been proposed initially in [10] for $p = 1$ and extended to arbitrary values for $p$ in [7]. The work [7] also contains an in-depth discussion of the $d^p$ measure in terms of (among others) applicability and computability. The work [7] also defines probabilistic satisfaction via $P(\psi \mid \phi) = d$ which requires a more careful treatment of the case $P(\phi) = 0$ and the necessity of introducing infinitesimal inconsistency values. However, in [7] no evaluation of the $d^p$-measure in terms of rationality postulates is given.

The work [1] also investigates the problem of reasoning in inconsistent probabilistic knowledge bases. There, reasoning based on the principle of maximum entropy is extended to be applicable on inconsistent knowledge bases. By doing so one eliminates the need for restoring consistency. Furthermore, [1] also proposes a continuous inconsistency measure which rests on the notion of *candidacy functions*, a "fuzzy" extension of probability functions. A thorough comparison of the measure of [1] with our approach is outside the scope of this paper but we refer to [11] for a comparison with the $d^1$-measure. However, note that the measure of [1] does not satisfy *super-additivity*.

In [9] another continuous inconsistency measure for probabilistic conditional logic is proposed that is not based on the $p$-norm distance but on *generalized divergence* which is a specific distance for probability functions. However, no technical results and no evaluation is given in [9].

## 7 Summary

In this paper we investigated continuous inconsistency measures for probabilistic conditional logic. We built on previous work and introduced several novel rationality postulates for inconsistency measurement that addressed the behavior of inconsistency measures with respect to continuity. It turned out that our measures satisfy most of the desired properties and, in particular, the compound measure also satisfies *penalty*.

The $d^1$-measure has already been implemented within the Tweety library for artificial intelligence[2] and future work includes implementation of the other measures. This will enable us to evaluate the behavior of the measures in more depth.

## References


1. Daniel, L.: Paraconsistent Probabilistic Reasoning. Ph.D. thesis, L'École Nationale Supérieure des Mines de Paris (2009)
2. Hansson, S.O.: A Textbook of Belief Dynamics: Theory Change and Database Updating. Springer-Verlag (1999)
3. Hunter, A., Konieczny, S.: Measuring Inconsistency through Minimal Inconsistent Sets. In: Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning. pp. 358–366. AAAI Press (2008)
4. Hunter, A., Konieczny, S.: On the Measure of Conflicts: Shapley Inconsistency Values. Artificial Intelligence 174(14), 1007–1026 (2010)
5. Kern-Isberner, G.: Conditionals in Nonmonotonic Reasoning and Belief Revision. No. 2087 in Lecture Notes in Computer Science, Springer-Verlag (2001)
6. Paris, J.B.: The Uncertain Reasoner's Companion – A Mathematical Perspective. Cambridge University Press (1994)
7. Picado-Muiño, D.: Measuring and Repairing Inconsistency in Probabilistic Knowledge Bases. International Journal of Approximate Reasoning (2011), to appear
8. Reiter, R.: A Logic for Default Reasoning. Artificial Intelligence 13(1–2), 81–132 (1980)
9. Rödder, W., Xu, L.: Elimination of Inconsistent Knowledge in the Probabilistic Expertsystem-Shell SPIRIT (in German). In: Operations Research Proceedings 2000. pp. 260–265. Springer-Verlag (2001)
10. Thimm, M.: Measuring Inconsistency in Probabilistic Knowledge Bases. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (2009)
11. Thimm, M.: Probabilistic Reasoning with Incomplete and Inconsistent Beliefs. Ph.D. thesis, Technische Universität Dortmund, Germany (2011), submitted


[2] http://sourceforge.net/projects/tweety/

# Learning Scenarios under Relational Probabilistic Semantics and ME Reasoning

Marc Finthammer and Nico Potyka

Dept. of Computer Science, FernUniversität in Hagen

**Abstract.** We present a learning scenario for relational learning which takes statistics on a population as well as uncertainty on individuals into account. It can be considered as an extension of a propositional maximum entropy (ME) framework which regards learning as the inverse process to inference. We start with the introduction of relational probabilistic conditional semantics and ME inference. Then we give a brief review of how learning can be performed in a propositional probabilistic framework involving ME reasoning. We use this propositional scenario to motivate the relational learning scenario and illustrate it by some examples. Finally we compare our relational learning scenario to some popular statistical relational learning approaches (MLNs, BLP, PRMs).

## 1 Introduction

In the field of probabilistic knowledge representation two important areas are inference and learning. Inference is concerned with inferring knowledge from a knowledge base, learning with learning knowledge bases from observed data. In recent years propositional approaches like Bayesian networks and Markov networks have been extended to the relational case using methods from the field of inductive logic programming. Their learning methods aim at general information, without taking uncommonly individuals into account.

The propositional framework of maximum entropy (ME) uses a set of probabilistic rules, so-called conditionals, as a knowledge base. A ME-inference function takes a set of conditionals and determines a probability distribution which satisfies all conditionals and has maximum entropy. In this way there is no more information added than needed. In [4] a learning algorithm is introduced, taking a probability distribution and calculating a set of conditionals. ME-inference applied to this set roughly calculates the original distribution again. In this way learning can be considered as inverse to inference.

ME-inference has been extended to the relational case already [6]. In this paper the relational learning problem shall be considered. The presented extension to the relational case not only allows the analysis of general information, but also of exceptional individual information.

In Section 2 the basic principles and maximum entropy inference methods are introduced. Firstly propositional inference is considered, then a brief overview of relational extensions is given. Section 3 is concerned with with learning. Again

firstly the propositional case is considered, then a concept for the relational extension is developed and illustrated by examples. Subsequently the propositional and relational cases are compared with respect to expressiveness and complexity. In Section 4 three relational extensions of graphical propositional frameworks are considered and subsequently compared to the maximum entropy approach. The paper concludes with a short outlook to future work.

## 2 Relational Probabilistic Semantics and Reasoning

Before considering the relational case, we give a brief overview of the propositional case to explain the basic concepts underlying both approaches. Ignoring formal stringency, we denote corresponding concepts by the same identifiers. It should be clear from the context what framework is considered.

### 2.1 ME-Optimal Reasoning in a Propositional Framework

Let $\Sigma = \{v_1, \ldots, v_n\}$ be a set of $n$ binary propositional variables and let $\mathcal{L}$ be a propositional language consisting of the propositional formulas over $\Sigma$. Thus for all variables $v \in \Sigma$ it holds $v \in \mathcal{L}$, and if $A, B \in \mathcal{L}$ then $\neg A, A \wedge B, A \vee B \in \mathcal{L}$. $v$ is called a positive, $\neg v$ a negative literal. A conjunction containing every variable as a positive or a negative literal exactly once is called a complete conjunction over $\Sigma$. Let $\Omega$ be the set of all $2^n$ complete conjunctions over $\Sigma$. These can be interpreted as the set of possible worlds (propositional interpretations) over $\mathcal{L}$. A probabilistic conditional (or rule) $(B \mid A)[x]$ consists of propositional formulas $A, B$ and a probability value $x$.

Semantics is given to conditionals by defining a probability distribution $P : \Omega \to [0, 1]$ assigning probabilities to worlds. Then the entailment relation $\models$ between a probability distribution $P$ and a probabilistic conditional $(B \mid A)[x]$ can be defined as follows.

$$P \models (B \mid A)[x] \quad \text{iff} \quad P(AB) = xP(A) \text{ and } P(A) > 0 \tag{1}$$

A set $\mathcal{R}$ of probabilistic conditionals is satisfied by $P$, respectively $P$ is a model for $\mathcal{R}$ iff P satisfies all conditionals in $\mathcal{R}$. In general there can be many probability distributions satisfying $\mathcal{R}$; a ME-inference function chooses the unique model with maximum entropy [4]. Entropy can be considered as a measure for the indifference within a probability distribution, it is defined as $H(P) := \sum_{\omega \in \Omega} P(\omega) \log P(\omega)$. Maximum entropy reasoning is performed by choosing the distribution

$$P^* = \mathrm{ME}_{\mathrm{prop}}(\mathcal{R}) := \arg \max_{P \models \mathcal{R}} H(P)$$

as a unique model for $\mathcal{R}$. So $P^*$ represents the incomplete (and uncertain) knowledge from $\mathcal{R}$ inductively completed to a full distribution by applying the ME-principle.

### 2.2 ME-Optimal Reasoning in a Relational Framework

We consider a set of first order predicates *Pred*, a set of variables *Var* and a finite set of constants *Const*. Let $p/k$ denote the predicate $p \in \mathit{Pred}$ with arity $k$. The set of atoms $\mathcal{A}$ over *Pred* with respect to *Var* and *Const* can be defined in the usual way by $p(t_1, \ldots, t_k) \in \mathcal{A}$ iff $p/k \in \mathit{Pred}, t_i \in (\mathit{Var} \cup \mathit{Const})$ for $1 \leq i \leq k$. For better readability we will usually omit the referring indices. Analogously a logical language $\mathcal{L}$ is defined as usual, that is, $A \in \mathcal{L}$ if $A \in \mathcal{A}$ and if $A, B \in \mathcal{L}$ then $\neg A, A \wedge B \in \mathcal{L}$. Let $A \vee B$ be the shorthand for $\neg(\neg A \wedge \neg B)$. Conditionals are of the form $\phi := (B \mid A)$, where $A, B \in \mathcal{L}$, $A$ is called the antecedence, $B$ the consequence of $\phi$. The set of all conditionals is denoted by $(\mathcal{L} \mid \mathcal{L})$. A grounding function gr maps terms and formulas to ground terms and ground formulas in the usual way.

Let $\mathcal{H}$ denote the Herbrand base, i.e. the set containing all ground atoms constructible from *Pred* and *Const*. Let $\mathrm{gr}(\mathcal{L}) := \bigcup_{A \in \mathcal{L}} \mathrm{gr}(A)$ be the set of all grounded formulas and analogously $\mathrm{gr}((\mathcal{L} \mid \mathcal{L})) := \bigcup_{\phi \in (\mathcal{L}|\mathcal{L})} \mathrm{gr}(\phi)$ be the set of all grounded conditionals. A Herbrand interpretation $\omega$ is a subset of the grounded predicates, that is $\omega \subseteq \mathcal{H}$. Using a closed world assumption, each ground atom $p_{\mathrm{gr}} \in \omega$ is interpreted as true and each missing ground atom is interpreted as false; in this way they are similar to the complete conjunctions in the propositional case. Let $\Omega$ denote the set of all possible worlds (respectively Herbrand interpretations), that is, $\Omega := \mathfrak{P}(\mathcal{H})$ (with $\mathfrak{P}$ denoting the power set) and consider a probability distribution $P : \Omega \to [0, 1]$. In this way grounded conditionals can be handled like in the propositional case. The following semantics describe how to handle conditionals containing variables.

**Semantics for Relational Probabilistic Conditionals** Whereas in propositional probabilistic logic a natural semantics is given by (1), in relational probabilistic logic the decision is not that easy due to its relational character. Whereas propositions can be considered as independent of each other, relational conditionals about both constants and variables may exist, that conflict with each other. For example the conditional $\mathit{Flies}(X)[0.9]$, $X \in \mathit{Var}$ is in conflict with the conditional $\mathit{Flies}(\mathit{penguin})[0.0]$, $\mathit{penguin} \in \mathit{Const}$, since $\mathit{Flies}(\mathit{penguin})[0.9]$ would be generated by grounding the former conditional. The problem can be handled by distinguishing subjective and statistical conditionals like stated in [3] for example. The following semantics tries to combine both approaches.

In [6] the averaging semantics and the aggregating semantics are introduced. Like above, let $A, B \in \mathcal{L}$ denote formulas over $\mathcal{L}$. The *averaging semantics* refers to the average of the possible groundings.

$$P \models_{\oslash} (B \mid A)[x] \quad \text{iff} \quad \frac{\sum_{(B_{\mathrm{gr}} \mid A_{\mathrm{gr}}) \in \mathrm{gr}((B|A))} P(B_{\mathrm{gr}} \mid A_{\mathrm{gr}})}{|\,\mathrm{gr}((B \mid A))|} = x$$

The *aggregating semantics*, however, is more a kind of decomposition of the conditional probability, in that it sums the probabilities of the conditional-satisfying

worlds in the numerator and the probability of the antecedence-satisfying worlds in the denominator.

$$P \models_{\odot} (B \mid A)[x] \quad \text{iff} \quad \frac{\sum_{A_{\mathrm{gr}} \in \mathrm{gr}(A), B_{\mathrm{gr}} \in \mathrm{gr}(B)} P(B_{\mathrm{gr}} \wedge A_{\mathrm{gr}})}{\sum_{A_{\mathrm{gr}} \in \mathrm{gr}(A)} P(A_{\mathrm{gr}})} = x$$

In [9] *grounding semantics* are introduced. The set of constants *Const* and the set of variables *Var* are partitioned by introducing *sorts*, in this way it is possible to restrict the grounding of general predicates to certain constants. Conditionals then are interpreted by interpreting all ground conditionals just like in the propositional case with respect to a special grounding operator $\mathcal{G}$.

$$P \models_{\mathcal{G}} (B \mid A)[x] \quad \text{iff} \quad P \models (B_{\mathrm{gr}} \mid A_{\mathrm{gr}})[x] \text{ for all } (B_{\mathrm{gr}} \mid A_{\mathrm{gr}}) \in \mathcal{G}((B \mid A))$$

That is, $P$ is a grounding-model of $(B \mid A)[x]$ iff $P$ is a propositional model for any ground instance given by $\mathcal{G}$.

**Reasoning under Maximum Entropy** In [6] it is stated that the averaging and aggregating semantics are compatible with ME-inference, that is, one obtains inference functions $\mathrm{ME}_{\oslash}$, $\mathrm{ME}_{\odot}$ like in the propositional case. Additionally it is shown that both semantics satisfy some desirable properties. Since the probability of each conditional is strictly determined, the inference function $\mathrm{ME}_{\mathcal{G}}$ can be directly derived from the propositional inference operator $\mathrm{ME}_{\mathrm{prop}}$, see [9] for details.
In the following, let $\mathrm{ME}_{\circ} \in \{\mathrm{ME}_{\oslash}, \mathrm{ME}_{\odot}, \mathrm{ME}_{\mathcal{G}}\}$ denote that inference function which determines the satisfying probability distribution with maximum entropy with respect to the given semantics:

$$\mathrm{ME}_{\circ}(\mathcal{R}) := \arg \max_{P \models_{\circ} \mathcal{R}} H(P)$$

## 3 Learning under Probabilistic Semantics

In this section, we will discuss some general aspects of a learning scenario involving the relational semantics and $\mathrm{ME}_{\circ}$ reasoning from Section 2.2. We will concentrate on a proper description of such a learning scenario, especially how the input data looks like. However, we will not deal with any practical learning algorithms. We start with a brief review of how learning can be performed in a propositional probabilistic framework involving ME reasoning. We will use this propositional scenario to motivate the relational one as a natural extension.

### 3.1 Propositional Learning Scenario

The approach taken in [5] defines learning as a process inverse to inductive knowledge representation. Therefore, one starts with given input data in terms of a probability distribution $P$, and the goal is to learn a set of probabilistic

rules $\mathcal{R}^*$, so that $P = \mathrm{ME}_{\mathrm{prop}}(\mathcal{R}^*)$ holds. Since the $\mathrm{ME}_{\mathrm{prop}}$ function is not injective, in general many solutions exist and one is interested in a solution $\mathcal{R}^* \in {\mathrm{ME}_{\mathrm{prop}}}^{-1}(P)$ which represents $P$ in a compact way. So the process of learning (or more precisely the process of conditional knowledge discovery) is to solve the inverse ME-problem by computing a preferably compact rule set $\mathcal{R}^*$ so that the given distribution $P$ is a ME-model for $\mathcal{R}^*$ (see Fig. 1). In [5], a practical learning algorithm and its implementation in the CondorCKD system is presented which can perform the calculation of $\mathcal{R}^*$.

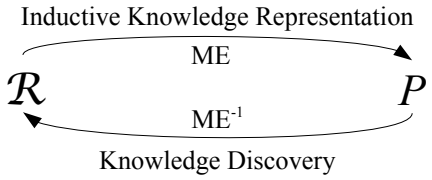


**Fig. 1.** Knowledge representation inverse to knowledge discovery

*Example 1.* The setting of this (fictional) example is a zoo where a population of monkeys lives. The monkeys have been observed for some time and each observation describes a monkey in terms of five binary attributes: *aggressive*, *hungry*, *male*, *nervous*, *sleepy*. So each observation can be depicted by a binary 5-tuple, i. e. by a complete conjunction over appropriate binary variables. Table 1 shows the observed frequency $freq(\omega)$ and corresponding probability $P(\omega)$ (i. e. relative frequency) of each tuple $\omega$.

The probability distribution $P_{\mathrm{T1}}$ from Table 1 serves as input data to learn a rule set $\mathcal{R}^*_{\mathrm{T1}}$ so that $P_{\mathrm{T1}} = \mathrm{ME}_{\mathrm{prop}}(\mathcal{R}^*_{\mathrm{T1}})$ holds. An algorithm which performs this kind of learning operation is described in [5]. Let the learned rule set $\mathcal{R}^*_{\mathrm{T1}}$ consist of the following rules:

$$\begin{array}{ll} (nervous & \mid hungry) \; [0.8] \\ (\neg hungry & \mid sleepy) \; [0.7] \\ (aggressive & \mid male \wedge hungry) \; [0.9] \end{array}$$

So this set $\mathcal{R}^*_{\mathrm{T1}}$ of generating rules compactly describes the observed distribution $P_{\mathrm{T1}}$ in an entropy-optimal way. Furthermore, the rules from $\mathcal{R}^*_{\mathrm{T1}}$ make certain connections between variables obvious that hold in $P_{\mathrm{T1}}$. These connections are not evident by just looking at the explicit representation of $P_{\mathrm{T1}}$.

### 3.2 Relational Learning Scenario

Considering one of the relational semantics from Section 2.2 and its corresponding entailment relation $\models_\circ$, together with the inference operator $\mathrm{ME}_\circ$, we can define relational knowledge discovery analogously to the propositional approach from Section 2.1, i. e. as an inverse process to $\mathrm{ME}_\circ$-based inductive knowledge representation.

| sleepy | nervous | male | hungry | aggressive | $freq(\omega)$ | $P(\omega)$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 65 | 0.0561 |
| 0 | 0 | 0 | 0 | 1 | 65 | 0.0561 |
| 0 | 0 | 0 | 1 | 0 | 7 | 0.0060 |
| 0 | 0 | 0 | 1 | 1 | 7 | 0.0060 |
| 0 | 0 | 1 | 0 | 0 | 65 | 0.0561 |
| 0 | 0 | 1 | 0 | 1 | 65 | 0.0561 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0.0009 |
| 0 | 0 | 1 | 1 | 1 | 9 | 0.0078 |
| 0 | 1 | 0 | 0 | 0 | 65 | 0.0561 |
| 0 | 1 | 0 | 0 | 1 | 65 | 0.0561 |
| 0 | 1 | 0 | 1 | 0 | 112 | 0.0966 |
| 0 | 1 | 0 | 1 | 1 | 112 | 0.0966 |
| 0 | 1 | 1 | 0 | 0 | 65 | 0.0561 |
| 0 | 1 | 1 | 0 | 1 | 65 | 0.0561 |
| 0 | 1 | 1 | 1 | 0 | 16 | 0.0138 |
| 0 | 1 | 1 | 1 | 1 | 140 | 0.1208 |
| 1 | 0 | 0 | 0 | 0 | 41 | 0.0354 |
| 1 | 0 | 0 | 0 | 1 | 41 | 0.0354 |
| 1 | 0 | 0 | 1 | 0 | 21 | 0.0181 |
| 1 | 0 | 0 | 1 | 1 | 21 | 0.0181 |
| 1 | 0 | 1 | 0 | 0 | 41 | 0.0354 |
| 1 | 0 | 1 | 0 | 1 | 41 | 0.0354 |
| 1 | 0 | 1 | 1 | 0 | 3 | 0.0026 |
| 1 | 0 | 1 | 1 | 1 | 26 | 0.0224 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 |

**Table 1.** Observed probability distribution $P_{\mathrm{T1}}$

When performing relational maximum entropy reasoning (as described in Section 2.2), we start with a set of relational probabilistic conditionals $\mathcal{R}$ and have to calculate the distribution $P^* = \mathrm{ME}_\circ(\mathcal{R})$. Consequently, the thereto inverse process of relational knowledge discovery (also see Fig. 1) is to determine a set of relational probabilistic conditionals $\mathcal{R}^*$, given a probability distribution $P$ over all possible worlds $\Omega$, so that $P = \mathrm{ME}_\circ(\mathcal{R}^*)$ holds. We will refer to this as "*relational ME-learning*" in the following. Next, we illustrate this learning scenario by another example.

*Example 2.* We take up again the setting from Example 1 and consider a population of monkeys. However, this time, we do not make observations of "anonymous" monkeys, but detailed observations of individual monkeys and their relationships. The population consists of three monkeys: *Andy*, *Bobby*, and *Charly*. Our observations focus on the feeding behavior of the monkeys, hence we note whether a monkey is *hungry* and which monkey *feeds* another one. Since the behavior of the monkeys is known to vary over time, we do not just take a single snapshot of the population, but make several observations over a longer period of time.

To describe our example setting in relational logic, we define the sets

$$Pred = \{Feeds/2, Hungry/1\} \quad \text{and} \quad Const = \{andy, bobby, charly\}$$

of predicates and constants[1]. An observation of the population states the truth value of every ground atom. Therefore, an observation $\omega$ is a Herbrand interpretation, i. e. possible world, from the set $\Omega$. Since there are 12 ground atoms, constructible from *Pred* and *Const*, there are $2^{12} = 4096$ possible worlds in $\Omega$.

For example, the observation $\omega_{332} = \{H(a), H(c), F(b, a), F(b, c)\}$ states that the monkeys Andy and Charly are hungry, Bobby is not hungry, Bobby feeds Andy and Charly, and that no other monkey feeds another one.

For each possible world $\omega$,[2] Table 2 shows its frequency $freq(\omega)$, i. e. how often $\omega$ has been observed, and its corresponding probability (i. e. relative frequency). It is important to note that (for short of notation) every world which is *not listed* explicitly in Table 2 has implicitly a frequency of 0. Hence, Table 2 only lists 37 of the overall 4096 worlds in $\Omega$.

Note that the predicate *Feeds* is meant to be irreflexive, i. e. a monkey never feeds itself. Therefore, each world where at least one of the ground atoms $Feeds(andy, andy)$, $Feeds(bobby, bobby)$, or $Feeds(charly, charly)$ is true has a mandatory frequency of 0. To keep things simple, we have omitted these ground atoms in the above considerations. Consequently, in each world $\omega$ from Table 2, these three ground atoms are implicitly false.

We use the probability distribution $P_{\mathrm{T2}}$ from Table 2 as input data for an (fictitious) algorithm which can perform the above described kind of knowledge discovery under aggregation semantics. That is, we want to determine a set of relational probabilistic rules $\mathcal{R}^*_{\mathrm{T2}}$, so that $P_{\mathrm{T2}} = \mathrm{ME}_{\odot}(\mathcal{R}^*_{\mathrm{T2}})$ holds. Let us assume that such a learning algorithm discovers a rule set $\mathcal{R}^*_{\mathrm{T2}}$ consisting of the following rules:[3]

$$
\begin{array}{lll}
r_1 : (Feeds(X, Y) & | \neg Hungry(X) \wedge Hungry(Y) & )[0.80] \\
r_2 : (Feeds(X, Y) & | Hungry(X) & )[0.0] \\
r_3 : (Feeds(X, Y) & | \neg Hungry(X) \wedge \neg Hungry(Y) & )[0.10] \\
r_4 : (Feeds(X, charly) & | \neg Hungry(X) & )[0.95] \\
r_5 : (Feeds(X, X) & | \top & )[0.0]
\end{array}
$$

So the first discovered rule $r_1$ states that is very likely that a not-hungry monkey feeds a hungry monkey. Rule $r_2$ expresses the certain knowledge that a hungry monkey never feeds another one. The next rule $r_3$ says that it is very improbable that a not-hungry monkey is fed by another one. Rule $r_4$ is different from the previous ones, because it makes a statement about an individual monkey: it is most probable that if a monkey is not hungry, it feeds the monkey Charly, i. e. regardless of whether Charly is hungry or not (perhaps because Charly is an

[1] We will use the abbreviations $F, H$ and $a, b, c$ for a compact notation of predicates and constants, respectively.

[2] Table 2 uses a compact notation for possible worlds, in which "1" ("0") indicates that the respective ground atom is (not) included in the Herbrand interpretation.

[3] Because currently no such learning algorithm is available, the input data $P_{\mathrm{T2}}$ has been manually constructed artificially. In fact, $\mathrm{ME}_{\odot}(\mathcal{R}^*_{\mathrm{T2}})$ slightly differs from $P_{\mathrm{T2}}$.

| $\omega$ | $H(a)$ | $H(b)$ | $H(c)$ | $F(a,b)$ | $F(a,c)$ | $F(b,a)$ | $F(b,c)$ | $F(c,a)$ | $F(c,b)$ | $Freq(\omega)$ | $P(\omega)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\omega_0$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0.006 |
| $\omega_4$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 15 | 0.019 |
| $\omega_{16}$ | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 15 | 0.019 |
| $\omega_{20}$ | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 30 | 0.038 |
| $\omega_{64}$ | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0.001 |
| $\omega_{68}$ | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 20 | 0.026 |
| $\omega_{80}$ | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 20 | 0.026 |
| $\omega_{84}$ | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 40 | 0.051 |
| $\omega_{128}$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0.006 |
| $\omega_{129}$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 10 | 0.013 |
| $\omega_{144}$ | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 10 | 0.013 |
| $\omega_{145}$ | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 20 | 0.026 |
| $\omega_{160}$ | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 10 | 0.013 |
| $\omega_{161}$ | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 20 | 0.026 |
| $\omega_{176}$ | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 20 | 0.026 |
| $\omega_{177}$ | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 40 | 0.051 |
| $\omega_{256}$ | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0.001 |
| $\omega_{258}$ | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 30 | 0.038 |
| $\omega_{260}$ | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 20 | 0.026 |
| $\omega_{262}$ | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 40 | 0.051 |
| $\omega_{264}$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0.006 |
| $\omega_{266}$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 10 | 0.013 |
| $\omega_{268}$ | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 10 | 0.013 |
| $\omega_{270}$ | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 20 | 0.026 |
| $\omega_{192}$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 10 | 0.013 |
| $\omega_{208}$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 20 | 0.026 |
| $\omega_{224}$ | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 20 | 0.026 |
| $\omega_{240}$ | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 40 | 0.051 |
| $\omega_{320}$ | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0.001 |
| $\omega_{324}$ | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 30 | 0.038 |
| $\omega_{328}$ | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 20 | 0.026 |
| $\omega_{332}$ | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 40 | 0.051 |
| $\omega_{384}$ | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0.006 |
| $\omega_{385}$ | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 20 | 0.026 |
| $\omega_{386}$ | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 20 | 0.026 |
| $\omega_{387}$ | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 40 | 0.051 |
| $\omega_{448}$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0.128 |

**Table 2.** Observed probability distribution $P_{\mathrm{T2}}$ (only listing worlds with a non-zero probability)

underfed baby monkey suffering from an eating disorder). Thus, $r_4$ describes a special case for Charly, because according to rules $r_1$ and $r_3$, one would have suspected that the feeding of Charly (by a not-hungry monkey) depends on whether Charly is hungry or not. The last rule (i. e. fact) $r_5$ just expresses that a monkey never feeds itself. What also becomes clear from these learned rules is that there seems to be no noticeable difference in the behavior of the two monkeys Andy and Bobby. Since none of the discovered rules make any special statement about them, these two monkeys are some kind of prototypical monkeys of the population.

As Example 2 has demonstrated, the relational learning scenario applies very well to situations where statistical information about several possible worlds is available and shall be processed in a way which also preserves information about exceptional individuals. This also fits to another example setting which we will have a brief look at: Consider a technical system consisting of several components

(the individuals in this case) which can be in different states (e. g. $Working(X)$, $Idle(X)$, $Suspended(X)$, $Defective(X)$, etc.) and which are related to each other (e. g. $WaitsFor(X, Y)$, $Powers(X, Y)$, etc.). Let the system be equipped with appropriate sensors to monitor and automatically protocol its complete current state (in terms of a Herbrand interpretation) at a fixed time interval. So a frequency distribution over possible worlds is generated. Using this distribution as input data for the *relational ME-learning* scenario, one might learn rules which describe general aspects of the system (e. g. $(WaitsFor(X, Y) \mid Idle(X) \wedge Working(X))[0.8]$). But one might also discover rules about correlations between distinct components (e. g. $Defective(machine5) \mid Powers(generator3, machine2)$ [0.9]) which could even make hidden problems of the system evident: e. g., a correlation might be discovered that if *generator3* powers *machine2*, then there is very often a defect of *machine5*; further investigations based on the discovered unusual correlation could unveil that the defect is due to an over-voltage which occurs only in this special situation.

### 3.3 Comparing Propositional and Relational Input

In the propositional learning scenario, the input data consists of a probability distribution over possible worlds. So we process statistical information about binary attribute vectors. Each attribute vector describes an *entity* by the truth values of all attributes and the (relative) frequency of each entity results from statistical observations. Although the input data from Example 1 was gathered by observing individuals, it just reflects a purely statistical view on anonymous individuals: each observed individual is just considered in terms of its attribute values and statistically counted as a corresponding entity.

However, in the relational learning scenario, the input data carries explicit information about distinct individuals identified by corresponding constants. The attributes of each individual are described by unary predicates and the corresponding ground atoms, respectively. Thus, if a relational scenario with solely unary predicates is considered, it resembles (roughly speaking) a propositional scenario but with distinct individuals. By introducing (at least) a single binary predicate, the difference in expressive power compared to the propositional case becomes much more obvious: since information about distinct individuals can be expressed, it is also possible to express information about relations between individuals. Obviously, this cannot be done in a propositional scenario, because there are no individuals (just entities carrying the statistical information about counted individuals). A possible world (Herbrand interpretation) in the relational scenario states which attributes and relations hold for each individual. By providing a frequency distribution over such possible worlds, the corresponding probability of a possible world $\omega$ can also be interpreted as a (subjective) degree of belief in the setting described by $\omega$.

At this point, it should already be mentioned that the relational learning approaches presented in Section 4 just processes (in general) a single Herbrand interpretation as input data. Roughly speaking, these approaches exploit the statistical information implicitly contained in a single (usually large) Herbrand

interpretation to learn generalized statements. We will discuss this in detail in Section 4.4.

Of course, one could simply map relational input data to propositional data by introducing a propositional variable for each ground atom. This way, it would be possible to use "relational" data within a propositional learning scenario. But by applying such a simple propositionalization, the semantical connection between the ground atoms will be lost, because the introduced propositional variables have no connections to their originating predicates and atoms (the name of a variable might indicate its corresponding ground atom but this is of no semantic relevance). So if the propositional learning algorithm from Section 3.1 was applied to the input data from Example 2 (by introducing a propositional variable for each ground atom, e. g. *feeds_andy_boby* for $Feeds(andy, bobby)$), the discovered rules would just make statements about ground atoms (after systematically replacing the "place holder" propositional variables with their corresponding ground atoms). That is, no rule could make a generalized statement using variables, but would just make statements about constants. The algorithm could also not make use of connections between certain ground atoms, e. g. between $Hungry(andy)$ and $Feeds(andy, bobby)$ both involving $andy$, because the algorithm would just "see" two distinct propositional variables *hungry_andy* and *feeds_andy_boby* without any connections in the first place.

### 3.4 Complexity of the Learning Scenarios

Comparing the propositional and the relational learning scenarios from the previous sections, both require statistical data in terms of a probability distribution $P$ over a set of possible worlds $\Omega$ as input data.

In the propositional case, there are $|\Omega_{prop}| = 2^{|\Sigma|}$ possible worlds, i. e. the size of the distribution $P_{prop}$ is exponential in the number of propositional variables. So for a "moderate" number of variables (which might still be enough for some interesting real world scenarios), a complete representation of $P_{prop}$ is large but still feasible.

On the contrary, in the relational case (with $\mathrm{ar}(pr)$ denoting arity of predicate $pr$), the Herbrand base consists of $|\mathcal{H}| = \sum_{pr \in Pred} |Const|^{\mathrm{ar}(pr)}$ ground atoms, resulting in $|\Omega_{rel}| = 2^{|\mathcal{H}|}$ possible worlds, i. e. the size of the distribution $P_{rel}$ is exponential in the number of predicates, constants, and especially in the predicates' arity. By using typed constants and predicates, the number of constructible ground atoms and therefore the size of $\mathcal{H}$ might be considerably reduced. Nevertheless, due to the exponential size in the number of ground atoms, the size of $\Omega_{rel}$ makes a complete representation $P_{rel}$ infeasible even for relatively small examples, as the following, just a "little bit" larger extension of Example 2 shows.

*Example 3.* Continuing Example 2, let the monkey population we observe this time consist of 10 monkeys. We extend our observations by the remaining four attributes from Example 1 and by four more relations between monkeys, leading to

the predicates[4] $Pred = \{hungry/1, sleepy/1, nervous/1, aggressive/1, male/1, feeds/2, attacks/2, delouses/2, playsWith/2, protects/2\}$. So we have 5 predicates with $\mathrm{ar}(pr) = 1$, 5 predicates with $\mathrm{ar}(pr) = 2$, and $|Const| = 10$ constants, resulting in $|\mathcal{H}| = 5 \cdot 10 + 5 \cdot 10^2 = 550$ ground atoms. Therefore, there are $|\Omega_{rel}| = 2^{550}$ possible worlds.

Regarding the size of $\Omega_{rel}$, it must be considered that in many example settings a large portion of possible worlds have a frequency of zero, so that the set of non-null worlds is only a small fraction of $\Omega_{rel}$. This is due to the fact that even though the number of possible worlds, i. e. potentially different observations, is theoretically enormous, the number of really observable worlds will probably be of feasible size, since many attributes and relations and the corresponding dependencies have a rather static character. Otherwise any observation will probably be represented by another world, in this case a more coarse-grained analysis of the data may be appropriate.

## 4 Other Probabilistic Relational Learning Approaches

At this point, we will have a look at some well-known approaches of statistical relational learning and especially what kind of relational input data these approaches process.

### 4.1 Markov Logic Networks

Markov logic [11] combines first-order logic with Markov networks [10] to establish a framework which allows the handling of a wide range of tasks from the area of statistical relational learning. The syntax of Markov logic in general conforms with first-order logic, but additionally each logic formula has assigned a real-valued weight value. The semantics of a set of Markov logic formulas is determined by a probability distribution over possible worlds. In contrast to classical logic, Markov logic formulas are not handled as hard constraints, instead each formula is softened depending on its weight. Thus, a possible world may still have a positive probability, although it violates a formula of the knowledge base. The weight of a formula determines its strength, i. e. how much the formula influences the probability of a satisfying world in contrast to a violating world.

A *Markov logic network (MLN)* $L$ is a finite set $L = \{(F_1, w_1), \ldots, (F_n, w_n)\}$ of pairs $(F_i, w_i)$ with a first-order logic formula $F_i$ and a real value $w_i$, its *weight*. Together with a set of constants $C$ it defines a *Markov network* $M_{L,C}$: The network contains a node for each constructible ground atom and an edge between two nodes iff corresponding ground atoms appear together in at least one grounding of a formula. For each possible grounding of each formula $F_i$, $M_{L,C}$ contains one binary feature (function) which is weighted by $w_i$.

[4] Note that the predicate *male* is obviously an attribute which has a fixed truth value for each individual (i. e. this truth value holds in every possible world). Therefore, it would be desirable, if such a predicate could be incorporated in some terms of background knowledge to remove this redundant information from each observation.

Therefore, an MLN defines a template for constructing *ground Markov networks*. That is, for a different set $C'$ of constants, a different ground Markov network $M_{L,C'}$ results from $L$, which may vary in size but whose general structure is quite similar (e. g. the groundings of a formula $F_i$ have the weight $w_i$ in any ground Markov network of $L$).

Let $n_i(\omega)$ denote the number of true groundings of a formula $F_i$ concerning a possible world $\omega$. A ground Markov network $M_{L,C}$ determines a probability distribution $P_{M_{L,C}}$ over possible worlds $\omega \in \Omega$ via the log-linear model [11]

$$P_{M_{L,C}}(\omega) = \frac{1}{Z} \exp\left(\sum_{(F_i,w_i)\in L} w_i n_i(\omega)\right) \tag{2}$$

using the normalization factor $Z = \sum_{\omega\in\Omega} \exp\left(\sum_{(F_i,w_i)\in L} w_i n_i(\omega)\right)$.

Markov logic allows probabilistic inference by calculating the conditional probability of a formula $B$ (the query) given a formula $A$ (the evidence).

Regarding Markov logic networks, two kinds of learning are to be distinguished: parameter (i. e. weight) learning and structure learning. When performing weight learning, the logical formulas of an MLN (i. e. its structure) are already known. The goal is to learn the missing weights (the parameters) for these formulas. Therefore, a (generative) weight learning algorithm tries to calculate weight values which maximize the log-likelihood of the input data. If it is already known at learn time which predicates will be queried and which ones will serve as evidence (e. g. in a collective classification task), it is more efficient to apply an algorithm which performs discriminative weight learning by maximizing the conditional log-likelihood of the query atoms given the evidence atoms of the input data. A MLN structure learning algorithm can either start with an initial set of MLN formulas or with an empty set. Then the algorithm tries to improve the log-likelihood of the input data by adding additional formulas (with appropriate weights) to the current MLN.

For both kinds of MLN learning, the input data is a set of ground atoms – in this context called *relational database* – which is considered under a closed world assumption. Therefore, the whole input data consists of *one* Herbrand interpretation, i. e. one possible world, $\omega_{input}$. As mentioned before, a MLN learning algorithm tries to find an MLN which maximizes the log-likelihood of this Herbrand interpretation $\omega_{input}$, i. e. the log-likelihood of equation 2 for a fixed $\omega_{input}$ and variable parameters $w_i$ (in case of weight learning) or variable parameter $L$ (in case of structure learning).

### 4.2 Bayesian Logic Programs

Bayesian Logic Programs combine Bayesian networks with first order logic [7]. First order dependencies are represented as a logical program, the contained clauses are grounded and used to construct a classical Bayesian network.

Bayesian clauses are of the form $c := A|A_1, \ldots, A_l$, where $A, A_1, \ldots, A_l \in \mathcal{A}$. The body and the head of a clause are denoted by $\text{body}(c) := \{A_1, \ldots, A_l\}$ and $\text{head}(c) := A$, corresponding to the antecedence and consequence of conditionals respectively. A Bayesian network can be described by a triple $\mathcal{B} = (\mathcal{C}, CPD, CR)$,

where $\mathcal{C}$ is a set of Bayesian clauses, *CPD* a set of conditional probability distributions and *CR* a set of combining rules. For any Bayesian clause $c$ *CPD* contains a conditional probability distribution $\mathrm{cpd}_c$, and for any Bayesian predicate $p$ *CR* contains a combining rule $\mathrm{cr}_p$. The combining rules are used to combine conditional probability distributions for the same variable, e.g. $P(A|B)$ and $P(A|C)$ can be combined to a new conditional probability distribution $P(A|BC)$. Possible combining rules are e. g. *noisy-or* and *average* [7].

Any Bayesian clause $c$ induces a set of grounded Bayesian clauses $\mathrm{gr}(c)$ by substituting any appearing variable by any possible constant, similar to the definition in Section 2. Any predicate appearing in $\bigcup_{c\in\mathcal{C}} \mathrm{gr}(c)$ then becomes a node in the corresponding Bayesian network. There is an edge from the grounded predicate $p_1^g$ to the grounded predicate $p_2^g$ iff there exists a grounded conditional $c$ with $\mathrm{head}(c) = p_2^g$ and $p_1^g \in \mathrm{body}(c)$. If the induced dependency graph is nonempty, acyclic and any node is influenced by a finite set of other nodes, it forms a Bayesian network. It factorizes in the usual way [7], that is $P(v_1, \ldots, v_{|\mathcal{V}|}) = \prod_{v\in\mathcal{V}} P(v|\,\mathrm{pa}(v))$. Thereby $\mathcal{V}$ is the set of nodes of the Bayesian network, that is, the grounded predicates, and $\mathrm{pa}(v)$ denotes the parent nodes of $v$. If $\mathrm{pa}(v)$ is induced by a single grounded clause $c$, then the necessary conditional probability distribution $\mathrm{cpd}_c$ is already contained in *CPD*. If there is more than one parent clause, the corresponding conditional probability distributions are aggregated using the corresponding combining rule $\mathrm{cr}_v$ from *CR*. A Bayesian logic program satisfying the above conditions of nonemptiness, acyclicity and finite influence is called well-defined.

Similar to Markov logic networks the learning problem can be distinguished in parameter learning and structure learning. Parameter learning refers to learning the conditional probability distributions; here the same approaches as for classical Bayesian networks can be adopted. Structure learning is more complicated since in this case not only the parameters but also the structure is unknown. The structure of the Bayesian network is induced by the clauses, so structure learning refers to learning the clauses respectively the logical program describing the problem. In [7] a greedy algorithm solving the problem is described. It starts with an initial well-defined logical program $\mathcal{C}$, that is determined using techniques from the ILP engine CLAUDIEN [1]. The clauses in $\mathcal{C}$ then are modified by refinement operators with respect to validity with respect to the learning data, acyclicity of the induced Bayesian network and a scoring function evaluating the quality of the current solution until no further improvement is obtained. The learning data is given by a set of data cases $\mathcal{D} = \{D_1, \ldots, D_l\}$. Each $D \in \mathcal{D}$ is a set of grounded predicates, similar to Herbrand interpretations, but additionally atoms of unknown truth value are possible.

### 4.3 Probabilistic Relational Models

Probabilistic Relational Models [2] show strong structural similarities to Bayesian Logic Programs, therefore we restrict ourselves to carve out the differences. Instead of a logical, a rather database-oriented framework is considered to induce

a dependency-graph. Their main parts are classes with their attributes and reference slots, corresponding to tables with their attributes and foreign keys in database terminology. Among the attributes, directed dependencies may exist, within a class as well as among classes. To each attribute a conditional probability distribution with respect to its parents is assigned. Similar to Bayesian Logic Programs an instantiated dependency graph can be generated, establishing a Bayesian network if it is acyclic. Indeed Probabilistic Relational Models presuppose acyclicity; the joint probability distribution over all variables factorizes then in the usual way. The problem of multiple conditional probability distributions to the same attribute, is overcome by database principles again, that is, the conflicts are resolved using aggregation functions like average or minimum.

As the relational framework suggests, Probabilistic Relational Models are able to learn from relational databases immediately. However, it is also possible to adapt the formalism to a single Herbrand interpretation using the closed world assumption [2]. The learning problem can be separated into parameter and structure learning again. Similar to Bayesian Logic Programs for parameter learning again Bayesian approaches are used. For structure learning a sophisticated greedy hill-climbing algorithm has been developed. Similar to Bayesian Logic Programs the treatment of incomplete data has been considered [8].

### 4.4 Comparison

What the three presented approaches have in common is an underlying graphical framework and a statistical relational semantics. Markov logic networks are based on Markov networks, Bayesian logic programs and Probabilistic relational models on Bayesian networks. Roughly speaking the basic learning concept is that each predicate (e.g. an attribute or a relation) of an individual depends only on some other predicates. Which predicates affect another predicate is independent of the individual. Hence it is sufficient to consider a single Herbrand interpretation to detect these dependencies. So one can roughly say that these approaches detect common dependencies between the individuals of a single world, that are used to form independent factors that altogether form a joint distribution. This joint distribution can be used to evaluate the quality of the factors to control a learning algorithm.

The relational ME-approach considered in this paper aims at detecting not only general dependencies, but also the individual dependencies. In particular, not only partial probability distributions but also more specific rules should be learned. Common rules can be combined to general rules using one of the three introduced semantics (averaging, aggregating and grounding). Seldom rules can be used to detect outliers, that can be further investigated. To analyze individuals in detail not one but several Herbrand interpretations, i.e. several observations, have to be taken into account. In this way not only the uncertain behavior of groups but simultaneously the uncertain behavior of the individual can be modeled.

## 5 Conclusion and Future Work

In this paper we presented a novel concept for relational learning and considered possible learning scenarios. Instead of considering deterministic individuals in a single Herbrand interpretation as in more popular approaches, the uncertainty of the individual is taken into account by considering probabilities on Herbrand interpretations. Of course, the gain in expressiveness is attended by a gain in complexity, but this is not as obstructive as a merely theoretical analysis indicates. This is due to the fact that in practical learning situations the size of possible outcomes can be significantly smaller than the number of theoretically possible worlds.

Nevertheless the presented learning task is currently merely conceptual. So in future work, we plan to develop a learning algorithm which can perform this learning task. For that purpose, we will investigate if (or to what extend) it can use certain concepts of the propositional learning algorithm from [5].

## References


1. De Raedt, L., Dehaspe, L.: Clausal discovery. Machine Learning 26(2–3), 99–146 (1997)
2. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning probabilistic relational models. In: IJCAI. pp. 1300–1309 (1999)
3. Friedman, N., Halpern, J.Y., Koller, D.: First-order conditional logic revisited. In: Proceedings of the thirteenth national conference on Artificial intelligence - Volume 2. pp. 1305–1312. AAAI'96, AAAI Press (1996)
4. Kern-Isberner, G.: Conditionals in nonmonotonic reasoning and belief revision. Springer, Lecture Notes in Artificial Intelligence LNAI 2087 (2001)
5. Kern-Isberner, G., Fisseler, J.: Knowledge discovery by reversing inductive knowledge representation. In: Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning, KR-2004 (2004)
6. Kern-Isberner, G., Thimm, M.: Novel semantical approaches to relational probabilistic conditionals. In: Lin, F., Sattler, U., Truszczyński, M. (eds.) Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR'10). pp. 382–392. AAAI Press (May 2010)
7. Kersting, K., De Raedt, L.: Basic principles of learning bayesian logic programs. Tech. Rep. 174, Institute for Computer Science, University of Freiburg, Germany (June 2002)
8. Li, X.L., Zhou, Z.H.: Structure learning of probabilistic relational models from incomplete relational data. In: Proceedings of the 18th European conference on Machine Learning. pp. 214–225. ECML '07, Springer-Verlag, Berlin, Heidelberg (2007)
9. Loh, S., Thimm, M., Kern-Isberner, G.: On the problem of grounding a relational probabilistic conditional knowledge base. In: Proceedings of the 14th International Workshop on Non-Monotonic Reasoning (NMR'10). Toronto, Canada (May 2010)
10. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1998)
11. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning 62(1–2), 107–136 (2006)

# Statistical Relational Learning in Dynamic Environments – An Agent-Based Approach to Dynamic Pathfinding Using Bayesian Logic Networks and ProbCog

Daan Apeldoorn

Dept. of Computer Science, FernUniversität in Hagen,
58084 Hagen, Germany
Daan.Apeldoorn@FernUni-Hagen.de

**Abstract.** Bayesian Logic Networks support statistical relational learning by combining conditional probability structures and distributions with evident knowledge and logical rules. This paper shows how these concepts can be applied to create an autonomously acting, self-learning agent. For this purpose, a pathfinding scenario in a dynamic environment will be introduced as an example: The agent will learn conditional probabilities by moving through a simulated traffic environment. Logical rules are implemented prior to the learning process to predetermine the agent's behavior in specific situations. The learned behavior can easily be comprehended after the learning process by inspecting the conditional probability network with the learned probability distributions. The implementation of the agent's Bayesian Logic Network is realized with ProbCog (an open-source software suite for statistical relational learning developed at the Technische Universität München), which is integrated into the external simulation application.

**Keywords:** Agent, Bayesian Logic Network, Dynamic Pathfinding, ProbCog, Simulation, Statistical Relational Learning.

## 1 Introduction

Creating agents which act adaptively in dynamic environments and improve their behavior by learning autonomously is an important application of artificial intelligence approaches. Representing an agent's model and the learned knowledge in an expressive and comprehensive way poses additional challenges. Statistical relational learning approaches combine logical inference with concepts of probability theory and machine learning [5] and can be useful to fulfill the introductory requirements.

In this paper, Bayesian Logic Networks [3], which support statistical relational learning, are used to implement an agent with the qualities described above. Compared to other self-learning techniques (Reinforcement Learning, for instance), the agent's learned knowledge can be easily inspected and comprehended after the

learning process. In addition, it is possible to inject logical rules selectively: The learning process can be supported with obvious correlations (which thus do not need to be learned) and certain behavior can be guaranteed.

As a test scenario, a simulated traffic environment with dynamic weather conditions is constructed in which the agent learns autonomously to optimize its navigation strategy to find the most adequate route from a starting point to a destination.

Section 2 starts with a short introduction to Bayesian Logic Networks according to [3]. Section 3 outlines the simulation environment and the agent's model with a Bayesian Logic Network using the ProbCog system [3]. Sections 4 and 5 will discuss the agent's resulting behavior and future work using Bayesian Logic Networks for agents in dynamic environments.

## 2. Bayesian Logic Networks

A Bayesian Network [4] is a representation of dependencies of conditional probability distributions as a directed acyclic graph. Nodes represent probability distributions and directed edges are used to describe the dependencies between these distributions. Fig. 1 shows a simple example of a Bayesian Network.

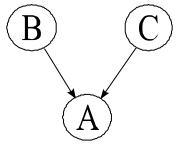


**Fig. 1.** Node A represents the conditional distribution P(A|B,C). Nodes B and C represent the distributions P(B) and P(C). Similar examples can be found in [2] et al.

Jain, Waldherr, and Beetz introduced Bayesian Logic Networks in [3]. Bayesian Logic Networks extend Bayesian Networks by concepts of logic to combine quantitative and qualitative methods. Thereby, a probabilistic knowledge base can be supported by logical rules and evident knowledge to ensure exact inference when possible.
According to [3], a Bayesian Logic Network is defined as a tuple

$$B = (D, F, L) . \quad (1)$$

The tuple D = (T, S, E, t) consists of a set T of the network's type declarations, a set S of function signatures, a set E of entities used for the type instantiations and a function t to assign the entities to their corresponding types. Types and their entities are used to describe the parameters and return values of the functions contained in S, where every function s in S is a tuple with the function name, a set of parameter types and the return type of the function. Logical predicates can be expressed as Boolean

functions and the type “Boolean” is always contained in T and its entities “True” and “False” are always contained in E.

The set F contains (conditional) probability distributions, where every distribution in F refers to a function in S. Therefore, every function can be considered a random variable that follows its corresponding distribution in F. The set F can be visually represented as a Bayesian Network, as described at the beginning of this section.

The set L consists of logical formulas over the functions (and predicates) contained in S.

By the concept of each node representing both a probability distribution in F and a function in S, quantitative as well as qualitative statements about functions (and predicates) are possible and evident knowledge can be added to the network in case function values are known for certain arguments.

A Bayesian Logic Network (as well as a conventional Bayesian Network) can be trained. The training data consists of value assignments to the functions and predicates of the network. The corresponding probability distributions are learned by counting relative frequencies of the assigned values: A learning algorithm counts how often every predicate evaluates to “True” or “False” and how often every function evaluates to certain values, according to the value assignments in the training data.

Quantitative and qualitative knowledge about functions and predicates can be inferred from a Bayesian Logic Network considering both the probability distributions and the logical rules.

## 3. Simulation Environment and Agent

In the first of the following subsections, the simulation environment and the problem to be solved by the agent are described. Afterwards, the second subsection explains the agent's model based on the concepts described in section 2 using the ProbCog system [3].

### 3.1 Description of the Environment

The environment is a street network with changing weather conditions. The street network consists of four different street types, namely “Freeway,” “Highway,” “Main Street,” and “Side Street,” which determine the agent's speed. The changing weather conditions also influence the speed as well as the probability of traffic jams on these different street types.

The agent has to find an adequate route through the street network to optimize the time needed to get from the starting point to the destination. Therefore, the agent has to decide which streets to take, depending on the type, the weather conditions, and the traffic jam probability. Fig. 2 shows a screen shot of the simulation environment.

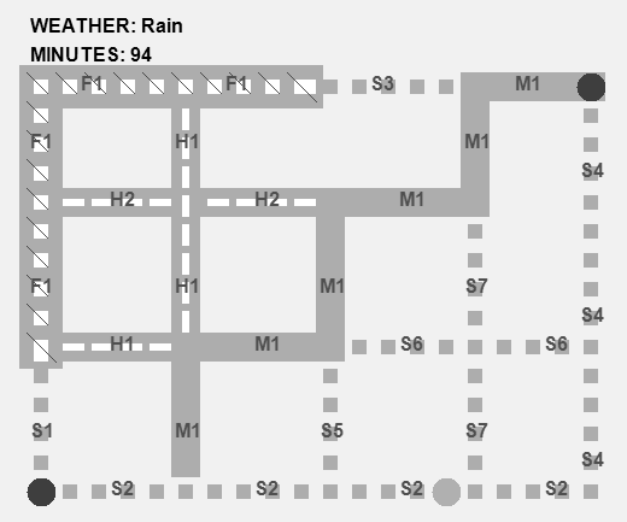


**Fig. 2.** The circle at the bottom left represents the starting point, the circle at the top right represents the destination. The remaining third circle displays the agent's position. The various street types are illustrated by different line styles. Streets with hatching indicate traffic jams.

### 3.2 Modeling the Agent

The agent is modeled with a Bayesian Logic Network using ProbCog, an open-source software suite for statistical relational learning, developed at the Technische Universität München [3].

**Types.** As a first step, the occurring street types and weather conditions have to be differentiated in order to form the agent's model. In addition, two further types are necessary to describe streets and the current situation of the environment. In ProbCog, this is achieved with the following type declarations:

Type Declarations Required to Implement the Agent as Bayesian Logic Network Using ProbCog.

```
type Weather;
guaranteed Weather Sun, Rain, Snow;

type StreetType;
guaranteed StreetType Freeway, Highway, MainStreet,
SideStreet;

type Street;

type Situation;
```

The keyword “guaranteed” ensures that ProbCog is informed about the existence of all named entities, even if the agent never comes across one or more of these entities while navigating through the environment.

**Random Variables.** In the second step the occurring random variables have to be declared as functions, which make use of the types previously declared. Random variables are needed for the environment's street type distribution and the weather conditions. In addition, there are random variables to characterize whether there is a traffic jam on a street in a given situation, whether a street is available in a given situation, whether a street is a detour or not, and whether it is wise to choose a street in a given situation. The latter random variable with the name “takeStreet” will be queried later to retrieve the inference results. In ProbCog, a random variable is introduced with the keyword “random” followed by the function name with its parameter types in parentheses. The following code shows the random variable declarations needed for the agent's model:

Random Variables Required by the Agent's Bayesian Logic Network.

```
random StreetType streetType(Street);
random Weather weather(Situation);
random Boolean trafficJam(Street,Situation);
random Boolean streetAvailable(Street);
random Boolean detour(Street);
random Boolean takeStreet(Street,Situation);
```

For reasons of simplification the random variables “streetAvailable” and “detour” do not depend on the current situation. This is not necessary, since in the environment scenario described above the topology of the streets is always the same while the simulation is running. Thus, this information can be provided without considering the current state of the environment. The information, if a street is available or whether it is a detour or not is provided as evident knowledge while the simulation is running (see section “Evident Knowledge”).

The conditional dependencies among the random variables can be modeled using ProbCog's integrated network editor[1]. Fig. 3 shows a screen shot of the complete network, created with the network editor.

1 The network editor can also be used after the agent's learning process to inspect the learned probabilities. For this purpose, a panel containing a probability table can be expanded for each node.

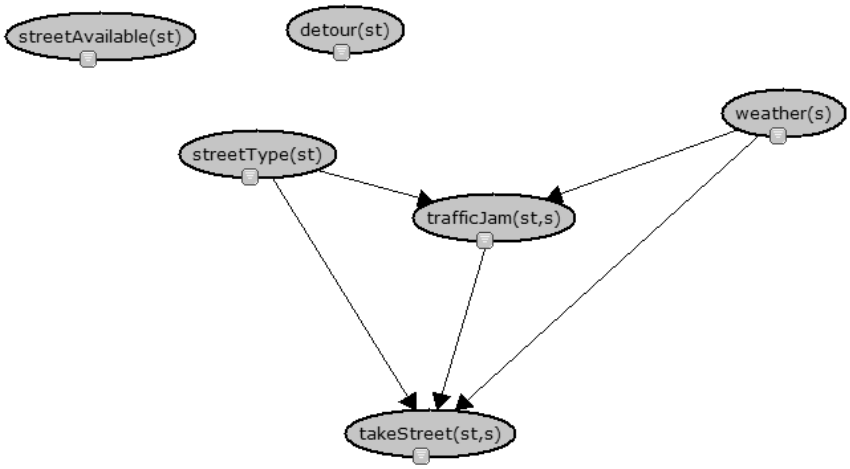


**Fig. 3.** The probability of traffic jams depends on a given street type and on the current weather conditions. Whether a street should be taken depends on the street type, the weather conditions and whether there is a traffic jam on the street in the current situation. The node "takeStreet(st,s)" is queried later for the inference results. The nodes "streetAvailable(st)" and "detour(st)" are provided for later use of logical rules and evident knowledge.

**Logic.** In the third step, the agent's model is completed by implementing the logical rules. By adding these rules, the agent's behavior can be predetermined for specific situations. This is useful in the beginning of the learning phase, when the agent has not gained enough knowledge yet. But it is also useful to guarantee a specific behavior in addition to what the agent will learn. Moreover, the implemented logic can support the agent's Bayesian Logic Network on correlations that are evident and thus do not need to be learned: The agent can make use of the logical rules instead of accessing the learned probability distributions.

In this model, only two logical rules are necessary: The first rule simply indicates that a street that is unavailable in the current situation cannot be taken by the agent. In ProbCog, this rule is implemented as follows:

Logical Rule to Determine that Only Currently Available Streets Can Be Taken.

```
(!streetAvailable(st)) => (!takeStreet(st,s)).
```

The second rule indicates that an available street, which represents a detour, will not be taken by the agent if it is affected by a traffic jam or if there is another street available in this situation without a traffic jam. This rule is implemented as follows:

Logical Rule to Drive Around Traffic Jams.

```
(streetAvailable(st) ^ detour(st)
^ streetAvailable(st2)
^ ((!detour(st2,s)) v trafficJam(st,s))
^ (!(st2 = st)))
=> (!takeStreet(st,s)).
```

The latter rule guarantees that detours are only accepted in order to drive around a traffic jam with the restriction that a traffic jam cannot be avoided by using a detour street which is itself affected by a traffic jam. This behavior (or a similar one) could also be learned and inferred from the probability distributions of the Bayesian Logic Network. But by implementing this rule, the agent will drive around traffic jams from the beginning of the learning phase, even if nearly nothing is known about the environment yet.

**Training Data.** While moving through the environment, the agent collects data about the environment's current traffic and weather conditions. At the beginning, the agent's behavior is based on random decisions to a larger extent. But while the simulation is running, it becomes more and more based on what the agent learns (see section “Diversification”). The collected information is continuously written to a file in the syntax used by ProbCog's learning module and this file is used as training data input for the Bayesian Logic Network: Every time the agent reaches its destination, ProbCog's learning module is called to learn the (conditional) probabilities for the Bayesian Logic Network from the collected training data. The following fragment represents one record of the training data collected by the agent while moving through the environment:

Example of a Training Data Record.

```
weather(s31) = Sun
streetType(S2_E) = SideStreet
trafficJam(S2_E,s31) = False
takeStreet(S2_E,s31) = True
```

As the example shows: In situation “s31,” the sun is shining and no traffic jam exists on street “S2_E” (side street 2 heading East). To determine whether a street is taken or not, the speed change is used as the only criterion in this scenario: If the agent is able to increase its speed after taking a certain street, it was preferable to take this street in this situation and “takeStreet” will be set to “True”. Otherwise, if the agent must decrease its speed after taking a certain street, “takeStreet” would be set to “False” for this street in this situation.[2]

2 A more planned behavior would be possible by using the difference of the time needed to reach the destination as criterion. But in this case, the streets chosen by the agent in the corresponding situations would have to be buffered until the destination is reached. After reaching the destination, it is determined if it was preferable to take these streets in these situations, depending on whether the agent needed more or less time to reach the destination.

The learning module works under the closed world assumption for Boolean random variables, so that all random variables are evaluated to “False” for all arguments for which no information about a variable is provided in the training data. This must be considered for the random variables “streetAvailable” and “detour”, since these two random variables are not present in the training data at all and thus will always have probabilities equal to 0 for “True” after the learning process. The real values for “streetAvailable” and “detour” are provided as evident facts immediately before the inference starts, and then are only used for evaluation of the logical rules. At this point, inconsistencies may occur between the evident knowledge provided and the learned knowledge in the probability distributions: If one of the variables is set to “True” as evident fact, this would represent a conflict in case the corresponding probability equals to 0. Such conflicts may lead to unpredictable inference behavior and must thus be avoided. This can be achieved by ensuring that the random variables “streetAvailable” and “detour” always possess probabilities greater than 0 for both “True” and “False” (the concrete probability values are not of importance since these random variables are only used for the logical rules as mentioned before). Therefore, the learned probability distributions for both random variables are simply replaced by the uniform distribution every time at the end of the agent's learning routine, subsequently to the call to ProbCog's learning module.

**Evident Knowledge.** On every crossroad, the Bayesian Logic Network is provided with evident knowledge before ProbCog's inference module is asked about which street to take. The evident knowledge provided consists of the current weather conditions, the available streets with their street types, which of the available streets is currently affected by a traffic jam, and which of the available streets is a detour to reach the destination. The evident knowledge is written to a file, which is then processed by the inference module. The evident knowledge's syntax is equal to the syntax of the training data file. As an example, a data record of evident knowledge is shown as follows.

Example of an Evident Knowledge Record.

```
weather(s) = Rain
streetType(F1_E) = Freeway
trafficJam(F1_E,s) = True
streetAvailable(F1_E) = True
streetType(H1_S) = Highway
trafficJam(H1_S,s) = False
streetAvailable(H1_S) = True
detour(H1_S) = True
```

In the situation described in the example, it is raining. The agent may choose between two streets: The freeway “F1_E” (freeway 1 heading East) and the highway “H1_S” (highway 1 heading South). On the freeway, there is currently a traffic jam, whereas the highway is free. The highway heading South is a detour, since it leads away from the destination point and thus taking this street would increase the distance to the destination. In the environment scenario described above it is very easy to

determine whether a street is a detour or not: According to the position of the starting point and the destination, streets heading South or East present detours and streets heading North or West do not present detours.[3]

**Diversification.** If some beneficial decisions have already been learned, it is still desirable for the agent to further explore the environment for possibly even better decisions. To achieve this, calling the inference module is diversified with a certain probability. This allows for the agent to replace a decision based on its learned knowledge with a random decision instead. To keep the agent's model simple and for improved traceability of the learning results, the diversification probability in the described scenario starts at 1.0 (random behavior) and is then discounted linearly while the simulation runs.[4] As a result, the agent is able to more and more exploit the learned knowledge and the agent's behavior approaches an optimized policy.

**Technical Realization.** The implementation of the agent is realized with the programming language C-mol[5] using Microsoft Visual Studio 2008 Express Edition and the .NET framework. Since ProbCog offers no native interface for this, the integration is realized by automatically writing the necessary data into the corresponding files used by ProbCog's learning and inference modules. The modules are then called as external processes. The standard out of the inference module is redirected to the main program and parsed to retrieve the results.

## 4. Results

The agent presented here shows a solid learning behavior. The learned probabilities for the weather conditions approximate quickly to the real probability distribution of the simulation environment. Acceptable precision of the probabilities is reached after only a few iterations. But the learned probabilities for traffic jam and whether or not a street should be taken are relatively low. This is based on the fact that ProbCog's learning module works under the closed world assumption (as described in section 3.2): All training data about traffic jams and whether or not a street should be taken is collected for the current situation only. Therefore, the corresponding random variables “trafficJam” and “takeStreet” are automatically evaluated to “False” for all other situations. This effect does not influence the inference to evaluate a street in a

3 In a more complex scenario, a detour could be determined for example with an A* search or a similar algorithm, if the topology of the map is known by the agent. If the agent does not know the (complete) topology, probabilistic or other approaches must be used instead.

4 More advanced strategies to control the diversification probability would be possible as well. For example, the diversification probability could depend on the average changes in the probability distribution of “takeStreet” (less changes meaning less diversification). This could lead to a more adaptive behavior to environment changes that appear later in the simulation.

5 C-mol (C++-based method-oriented language) is a programming language developed by Henrik Heimbürger and the author, where the methods of a program (instead of its classes) are the central development units [1].

certain situation, since the probabilities are correct in relation to each other. But nevertheless, the possibility to enable the closed world assumption for certain random variables (as possible in the inference module) might be a useful extension to ProbCog. Fig. 4 shows the agent's learning behavior with linear discounting of the diversification probability.

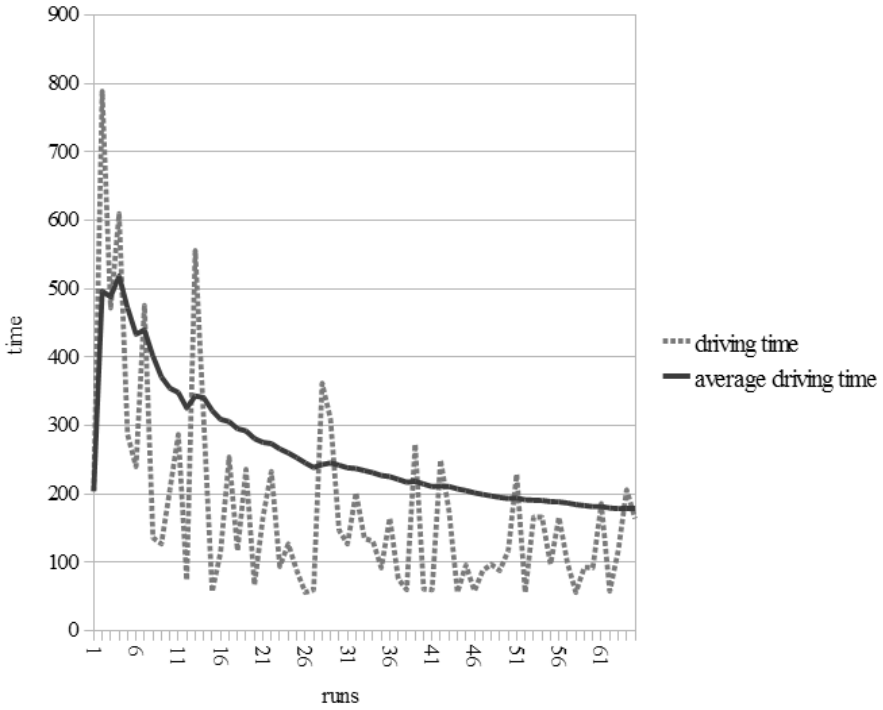


**Fig. 4.** The x-axis represents the number of simulation runs and the y-axis represents the time needed to reach the destination. The agent's required driving time (*dotted line*) and the average driving time (*continuous line*) are visualized. The diversification probability is discounted linearly and the agent's (average) driving time decreases disproportionately.

The agent's learning success does not directly depend on the scenario size, since the number of conditional probabilities that have to be learned by the agent remains constant for larger street networks. Nevertheless, approximating an adequate behavior could take longer in case larger regions of one street type exist, since the agent would need more time to come across and collect data about all different street types. The same applies to the relocation of the starting point and the destination. But in this case, the determination of detours can become more complex and must be adapted (see section 3.2, “Evident Knowledge”).

The integration of Bayesian Logic Networks via ProbCog into the external simulation environment is relatively easy, although there is no native interface for the programming language used. The well structured output of ProbCog's inference module can be easily parsed to obtain the results in the host application. But since the modules must be called as external processes for learning and for the inference queries, small delays are noticeable in the simulation every time ProbCog is called.

This might become a problem when using ProbCog in non-native environments where true real-time processing is required.

## 5. Conclusion and Future Work

As demonstrated in the previous sections, Bayesian Logic Networks are a useful concept to create agents for dynamic environments. Agents can be modeled by defining the random variables of the environment and their dependencies. Logical rules can be added to inject a specific behavior in addition to what an agent will learn. This may turn out to be very useful in practical applications, where adaptivity is needed but a specific behavior must be guaranteed in some situations. The resulting agent learns to optimize its behavior autonomously and is adaptive to the dynamics of the environment. The learned knowledge can easily be examined and comprehended, which represents an advantage in comparison with other self-learning approaches.

But even if the agent's model can be created easily, it would be desirable if the conditional probability network structure of the Bayesian Logic Network could also be learned from the agent's collected training data. This would not only simplify the creation process of agents even further. Moreover, it would allow to create agents for environments where nothing is known about the probabilistic model. By also learning the Bayesian Logic Network structure, the learned knowledge would still remain easy to examine and to comprehend. At the same time, predetermined behavior could still be supplied with logical rules.

## References


1. Apeldoorn, D., Heimbürger, H.: Method-oriented software development (MOSD) with the programming language C-mol – A new concept for more efficient development and implementation of software systems. In: Gesellschaft für Informatik (ed.), Informatiktage 2003: Fachwissenschaftlicher Informatik-Kongress 7. und 8. November 2003 im neuen Kloster Bad Schussenried, pp. 103--106. Konradin Verlagsgruppe, Grasbrunn (2004)
2. Beierle, C., Kern-Isberner, G.: Methoden der Wissensrepräsentation und -verarbeitung. FernUniversität in Hagen, Hagen (2008)
3. Jain, D., Waldherr, S., Beetz, M.: Bayesian Logic Networks. Technical Report IAS-2009-03. Retrieved from http://ias.in.tum.de/publications/pdf/jain09blns.pdf, Munich (2009)
4. Jensen, F. V.: An Introduction to Bayesian networks. UCL Press, London (1996)
5. de Raedt, L., Kersting, K.: Probabilistic Inductive Logic Programming. In: de Raedt, L., Frasconi, P., Kersting, K., Muggleton, S. (eds.) Probabilistic Inductive Logic Programming. Theory and Applications. LNAI, vol. 4911, pp. 1--27. Springer, Berlin, Heidelberg (2008)

# On Efficient Algorithms for Minimal ME-learning

Nico Potyka

Dept. of Computer Science, FernUniversität in Hagen

**Abstract.** In a knowledge representation sense, the principle of maximum entropy (ME) states that a set of probabilistic rules is best represented by that probability distribution satisfying all rules and possessing maximum entropy. Accordingly an ME-inference function takes a set of probabilistic rules and determines the ME-optimal distribution representing this rules. CondorCKD is an implementation of an algorithm that inverts ME-inference in a special sense and realizes in this way a learning approach, completing the logical ME-framework. However, it does not scale well for bigger problems. In this paper an alternative approach is considered. Instead of an algebraic inversion strategy, the problem is regarded as a combinatorial optimization problem. After a closer look on this problem a simple top-down alternative to CondorCKD is presented, achieving a significant performance gain.

## 1 Introduction

Probabilities are proven means to represent degrees of belief or statistical information, hence probabilistic frameworks like Bayesian networks [11] or Markov random fields [10] are popular for knowledge representation tasks. Whereas they are based on a graphical structure, in the maximum entropy (ME) framework a more symbolic formalism is used [8]. What they all have in common is that a complete probability distribution is induced by a compact knowledge base, i.e., a bayesian network, a Markov random field or a set of rules in case of the ME-framework. In each formalism there is an inference procedure, able to create a whole probability distribution from such a knowledge base. Also one may be interested in a convenient knowledge base for observed data, this task is called learning. The learning problem is well-examined for the mentioned graphical models, see e.g. [11].

In the ME-framework a set of rules, so-called conditionals, is used as knowledge base. In [8] an algorithm is developed that exploits a special structure induced by an ME-inference function to invert this function in a sense. More precisely, the algorithm takes a probability distribution and calculates a set of conditionals. ME-inference applied to this conditional set approximately calculates the original distribution again. In this way it realizes a well-founded learning procedure completing the ME-framework. It follows a bottom-up approach. Most special rules able to represent any possible dependency are successive shortened, with intention of decreasing redundany while preserving important information.

In [9] CondorCKD is presented, an implementation of the algorithm in the functional programming language Haskell. Even though good results for smaller problems can be obtained, runtime and learning performance lower with increasing problem complexity. This paper focuses on more efficient learning of minimal knowledge bases, thereto the problem will be reformulated as a combinatorial optimization problem. Then a top-down approach will be developed as an alternative to CondorCKD. Instead of decreasing the rule set as long as a correctness criteria is maintained, a most general set is adequately increased until the criteria is satisfied.

In Section 2 the basic principles are explained, i.e. the logical formalism and the maximum entropy framework. Subsequently in section 3 the minimal ME-learning problem is considered. Initially a reasonable upper bound for the size of a knowledge base is determined, then the problem is formulated as a combinatorial optimization problem. In Section 4 then a greedy top-down-approach is considered to solute the problem heuristically. After a short examination of probabilistic difference measures a beneficial problem separation is introduced. Subsequently a greedy top-down algorithm and experimental results for two implementations are presented. The paper concludes with a short discussion and ideas for further work.

## 2 Basic Principles

In this section the basic principles are introduced. Since the notation varies, it must be mentioned, that in the following the power set, i.e., the set of all subsets of a set $A$ is denoted by $2^A$.

### 2.1 Multi-valued Propositional Probabilistic Logic

In most cases learning data is not represented as a probability distribution, but as a set of data instances $\mathcal{D}$, e.g., tuples in a database or, more generally, a table of observations. This dataset $\mathcal{D}$ with its corresponding attributes induces a probability distribution $\mathcal{P}_\mathcal{D}$ due to the relative frequencies of different instances. Instances of the dataset are represented by words of a probabilistically logical language $\mathcal{L}$ over a finite set of $n$ multi-valued propositional variables $\mathcal{V} = \{V_1, \ldots, V_n\}$ representing the attributes. To each variable a finite domain $\mathrm{dom}\,(V_i) = \{v_{i1}, \ldots, v_{ik_i}\}$ of values is assigned. Then $\mathcal{L}$ consists of all multi-valued propositional formulas formed in the usual way by conjoining finitely many atoms by conjunction, disjunction and negation. For ease of notation the atom $(V = v)$, $V \in \mathcal{V}$, $v \in \mathrm{dom}(V)$, is sometimes abbreviated by $v$, $F \wedge G$ by $FG$ and $\neg F$ by $\overline{F}$. The classical truth values true and false are denoted by $\top$ and $\bot$. The conjunctions in $\Omega := \{\bigwedge_{i=1}^{n}(V_i = v_i) | v_i \in \mathrm{dom}(V_i) \text{ for all } 1 \leq i \leq n\}$, containing exactly one proposition to each variable, are called *complete conjunctions*.

Obviously each instance of the dataset can be represented by an $\omega \in \Omega$, hence they are also called *possible worlds*. The set of all complete conjunctions,

respectively the set of all possible worlds $\Omega$, corresponds to the elementary events of the considered probability distribution $\mathcal{P}$. Hence intuitively speaking $\mathcal{P} : \Omega \to \mathbb{R}$ assigns to each instance of the dataset its relative frequency. In a more logical way each complete conjunction $\omega \in \Omega$ also corresponds to a propositional interpretation $I_\omega$, since it assigns a truth value to each variable. Let $\mathrm{Mod}(F) := \{\omega \in \Omega | I_\omega \models F\}$ denote the set of all worlds corresponding to classical models of a formula $F \in \mathcal{L}$. Note that each formula $\mathcal{F}$ is equivalent to the disjunction of complete conjunctions corresponding to its classical models, i.e., $\mathcal{F} \equiv \bigvee_{\omega \in \mathrm{Mod}(\mathcal{F})} \omega$. In this way a consistent probability to each formula can be assigned as follows:

$$\mathcal{P}(\mathcal{F}) = \mathcal{P}(\bigvee_{\omega \in \mathrm{Mod}(\mathcal{F})} \omega) = \sum_{\omega \in \mathrm{Mod}(\mathcal{F})} \mathcal{P}(\omega). \tag{1}$$

Conjunctions containing at most one proposition per variable are called *elementary conjunctions*. Hence each complete conjunction is an elementary conjunction, too. Conjunctions $C$ with $\mathcal{P}(C) = 0$ are of special interest as will be shown later, they are called *nullconjunctions (with respect to $\mathcal{P}$)*. As usual the atom $v$ is called positive Literal, $\overline{v}$ negative literal. Since all variables are multi-valued, negative literals are of little importance in this paper. For a boolean variable $b$ this may be a little bit confusing. Note that the positive literal $(b = false)$ is equivalent to the negative literal $\neg(b = true)$.

**Conditionals:** Dependencies $A \rightsquigarrow B$ between formulas $A, B \in \mathcal{L}$ are represented by so-called *conditionals* $(B|A)[x]$. $B$ is called consequence, $A$ is called antecedence of the conditional. The reason for the notation becomes obvious, when assigning probabilistic semantics to conditionals as follows:

$$\mathcal{P} \models (B|A)[x] \qquad \text{iff} \qquad \mathcal{P}(A) > 0 \text{ and } \mathcal{P}(B|A) = x. \tag{2}$$

Thereby $\mathcal{P}(B|A) := \frac{\mathcal{P}(AB)}{\mathcal{P}(A)}$ is the conditional probability of B given A. The definition of the probabilistic model relation can be extended to sets of conditionals $\mathcal{R}$ naturally by $\mathcal{P} \models \mathcal{R}$ iff $\mathcal{P} \models (B|A)$ for all $(B|A) \in \mathcal{R}$. Let $(\mathcal{L}|\mathcal{L})$ denote the set of all probabilistic conditionals. Conditionals $(B|A)$ satisfying $\mathcal{P}(A) > 0$, are called *consistent (with respect do $\mathcal{P}$)* in the following.

Conditionals $(B|A)[x]$ are called *deterministic conditionals (with respect to $\mathcal{P}$)* if $x \in \{0, 1\}$ and *non-deterministic* otherwise. The consequence of *single-elementary conditionals*e consists of exactly one literal and their antecedence is a disjunction of elementary conjunctions not containing the consequence variable. In many cases conjunctions will be sufficiently expressive, discjunctions may become necessary when considering the negation of multi-valued variables. In this paper it is assumed that elementary conjunctions are sufficiently expressive. Let $(\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}}$ denote the conjunction-restricted set of single-elementary conditionals that are consistent with respect to $\mathcal{P}$. They can be perceived as a sort of multi-valued, probabilistic horn-clauses.

*Example 1.* Consider an expert system that shall learn some basic concepts about cars. For this purpose several cars are observed and evaluated with respect to their A(ge), P(erformance), F(uel type) and C(onsumption). The corresponding domains are $\mathrm{dom}(A) = \{new, used\}$, $\mathrm{dom}(P) = \{high, med, low\}$, $\mathrm{dom}(F) = \{petrol, diesel\}$ and $\mathrm{dom}(C) = \{high, med, low\}$. Hence the set of possible worlds $\Omega$ consists of all the possible configurations represented by complete conjunctions like $(A = new) \wedge (P = high) \wedge (F = petrol) \wedge (C = high)$. The two tables in Figure 1 show the observed cars, with their corresponding frequencies and relative frequencies interpreted as a probability distribution $\mathcal{P}$. Thereby every row corresponds to a complete conjunction. The nullconjunctions are not explicitly listed.

| A | P | F | C | frequency | $\mathcal{P}(\omega)$ |
|---|---|---|---|---|---|
| used | high | diesel | med | 12 | 0.012 |
| used | med | diesel | med | 12 | 0.012 |
| new | high | diesel | high | 13 | 0.013 |
| new | high | diesel | med | 13 | 0.013 |
| new | med | diesel | med | 13 | 0.013 |
| used | high | diesel | high | 18 | 0.018 |
| used | low | petrol | low | 54 | 0.054 |
| used | high | petrol | med | 54 | 0.054 |
| used | low | petrol | med | 54 | 0.054 |
| used | med | petrol | med | 54 | 0.054 |

| A | P | F | C | frequency | $\mathcal{P}(\omega)$ |
|---|---|---|---|---|---|
| new | high | petrol | high | 57 | 0.057 |
| new | high | petrol | med | 57 | 0.057 |
| new | low | petrol | med | 57 | 0.057 |
| new | med | petrol | med | 57 | 0.057 |
| new | low | petrol | low | 62 | 0.062 |
| used | low | diesel | low | 78 | 0.078 |
| used | low | diesel | med | 78 | 0.078 |
| used | high | petrol | high | 81 | 0.081 |
| new | low | diesel | med | 83 | 0.083 |
| new | low | diesel | low | 89 | 0.089 |

**Fig. 1.** Learning data for an expert system.

Now the application of a learning algorithm may yield the conditional $(C = high|P = high)[1]$, expressing that the observed cars with a high performance have also a high fuel consumption. Another conditional may be $(C = low|F = diesel)[0.8]$, expressing that observed cars consuming diesel often have a low fuel consumption.

### 2.2 The ME-Principle

In an information theoretical sense entropy can be seen as a measure for uncertainty inherent to a probability distribution $\mathcal{P}$. Given a set of conditionals $\mathcal{R}$ there can be many probability distributions satisfying $\mathcal{R}$. The principle of maximum entropy demands to choose that satisfying distribution possessing maximum entropy [7].

A more general notion is the *relative entropy*, which is a measure for the distance between two probability distributions. For probability distributions $\mathcal{P}_1$, $\mathcal{P}_2$ defined on the same discrete set of elementary events $\Omega$ it is defined as follows:

$$\mathrm{R}(\mathcal{P}_1, \mathcal{P}_2) := \sum_{\omega \in \Omega} \mathcal{P}_1(\omega) \cdot \log \frac{\mathcal{P}_1(\omega)}{\mathcal{P}_2(\omega)}.$$

By setting $0 \cdot \log \frac{0}{0} := 0$ and demanding $\mathcal{P}_2(\omega) = 0$ implies $\mathcal{P}_1(\omega) = 0$ it is well-defined. It is well-known that minimizing the relative entropy of a probability distribution $\mathcal{P}$ to the uniform distribution corresponds to maximizing the entropy of $\mathcal{P}$. To underline this duality between minimization of the relative entropy and maximization of the entropy, one often simply speaks of the *ME-Principle*.

**ME-inference:** Let $\mathcal{R} = \{(B_1|A_1)[x_1], \ldots, (B_m|A_m)[x_m]\}$ be a set of $m$ probabilistic conditionals. It turns out that the optimization problem of finding a probability distribution representing $\mathcal{R}$ while maximizing entropy leads to a special structure [8]. For every conditional $(B_i|A_i)[x_i]$ a function $\mathrm{f}_i$ is defined as follows:

$$\mathrm{f}_i(\omega) = \begin{cases} \alpha_i^{1-x_i}, \text{ if } I_\omega \models A_i B_i \\ \alpha_i^{-x_i}, \text{ if } I_\omega \models A_i \overline{B_i} \\ 1, \quad \text{if } I_\omega \models \overline{A_i}. \end{cases} \tag{3}$$

The $\alpha_i \in \mathbb{R}$ result from the solution of the optimization problem, see [8] for details. Hence $\mathrm{f}_i$ maps a world $\omega \in \Omega$ to a real number, dependent on the logical interpretation of $(B_i|A_i)$ with respect to the corresponding logical interpretation to $\omega$. Then the optimal solution is given as follows:

$$\mathcal{P}^*(\omega) = \mathrm{ME}(\mathcal{R})(\omega) = \alpha_0 \prod_{1 \leq i \leq m} \mathrm{f}_i(\omega). \tag{4}$$

Thereby $\alpha_0$ is just a normalizing constant ensuring $\sum_{\omega \in \Omega} \mathcal{P}^*(\omega) = 1$ so that $\mathcal{P}^*$ is a probability distribution. The *inference function* ME maps a set of conditionals $\mathcal{R}$ to the ME-optimal probability distribution representing $\mathcal{R}$.

*Example 2.* The probability distribution $\mathcal{P}$ in Figure 1 is generated by ME-inference applied to the condional set $\mathcal{R}$ shown in Figure 2, i.e., $ME(\mathcal{R}) = \mathcal{P}$. Let $\mathcal{R}'$ be the set of conditionals resulting from $\mathcal{R}$ by replacing the conditional $(C = high|P = high)[1.0]$ by the two conditionals $(C = low|P = high)[0.0]$ and $(C = med|P = high)[0.0]$. Then $\mathrm{ME}(\mathcal{R}') = \mathcal{P}$ also holds, hence ME is not injective.

$(C = low|F = diesel)[0.8]$
$(P = high|A = used)[0.2]$
$(P = low|A = new)[0.3]$

$(C = high|P = high)[1.0]$
$(C = low|P = low)[1.0]$

**Fig. 2.** Conditional set inducing car data.

**ME-learning:** In [8] it is proposed to consider ME-learning as a process being inverse to ME-inference. As example 2 demonstrates the ME-function is not injective, i.e., there can be more than one conditional set inducing the observed probability distribution. But defining two conditional sets as equivalent with

respect to ME-inference iff they induce the same probability distribution, it can be inverted to representatives of equivalence classes $[\mathcal{R}]_{ME} := \{\mathcal{R}' | \, \mathrm{ME}(\mathcal{R}') = \mathrm{ME}(\mathcal{R})\}$.

Based on [8] in [9] CondorCKD is presented. It takes a probability distribution $\mathcal{P}$ and tries to compute a knowledge base $\mathcal{R}$ to $\mathcal{P}$. It starts with the set of single-elementary conditionals of maximal length, the so-called *basic conditionals*. Numerical dependencies in $\mathcal{P}$ can be used to derive dependencies in the conditional set. Using these dependencies the basic conditional set is stepwise decreased by combining two conditionals to a shorter one or deleting conditionals. Unfortunately the derived dependencies cannot be guaranteed to be valid, unless $\mathcal{P}$ is a so-called *faithful representation* of $\mathcal{R}$, see [8] for details.

To cover unfaithful representations, CondorCKD uses several heuristics [9]. But with increasing number of variables there are either too many invalid shortenings of the basic conditional set or too many remaining conditionals. To avoid *invalid shortenings*, i.e., shortenings entailing loss of information, backtracking can be used [8]. A naive backtracking check could be implemented by infering a probability distribution from the shortened conditional set and comparing it to the original distribution. Nevertheless, a valid shortening at a certain point may not be optimal. That is, the information contained in the shortened conditional may be equivalently expressed by several other remaining conditionals. But after shortening the conditional all the other conditionals have to remain to conserve the information. Hence in this sense it is only a *locally valid shortening*. What is more, both ME-inference and the comparison of distributions may be computation-costly. Since CondorCKD starts with a number of single-elementary conditionals exponential in the number of variables, this is a serious problem, since a huge number of checks might be necessary.

## 3 Minimal ME-learning

In this section the problem of inverting ME-inference, resp. the ME-learning problem, is considered in detail. As mentioned above there is not necessarily a unique solution. Given a probability distribution $\mathcal{P}$ a set of conditionals $\mathcal{R}$ is searched, satisfying $\mathrm{ME}(\mathcal{R}) = \mathcal{P}$. Then $\mathcal{P}$ is called ME-representation of $\mathcal{R}$, $\mathcal{R}$ is called knowledge base to $\mathcal{P}$. The focus of this paper is on finding a knowledge base of minimal size to a given distribution $\mathcal{P}$, called *minimal ME-learning* in the following. After considering an upper bound for the number of conditionals, the problem is reformulated as a combinatorial optimization problem.

### 3.1 An upper Bound for a Knowledge Base

Consider a set of $n$ multi-valued variables $\mathcal{V} = \{V_1, \ldots, V_n\}$ with the corresponding logical language $\mathcal{L}$, the set of worlds $\Omega$, the probability distribution $\mathcal{P} : \Omega \to \mathbb{R}$ and the consistent conjunctive single-elementary conditionals $(\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}}$ as explained in Section 2.1. For ease of notation it is assumed that $|\operatorname{dom}(V)| = k$ for all $V \in \mathcal{V}$. Then the probability distribution $\mathcal{P}$ can be uniquely described

by $k^n$ real-valued variables $x_\omega$ corresponding to the probabilities of the possible worlds. Every single-elementary conditional $(v|A)[p] \in (\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}}$ induces an equation over some variables $x_\omega$ as follows. By definition (2) $\mathcal{P}$ is a model for $(v|A)[p]$ iff the following holds:

$$p = \mathcal{P}(v|A) = \frac{\mathcal{P}(Av)}{\mathcal{P}(A)} = \frac{\mathcal{P}(Av)}{\mathcal{P}(A\overline{v}) + \mathcal{P}(Av)}.$$

Hence $(1-p) \cdot \mathcal{P}(Av) - p \cdot \mathcal{P}(A\overline{v}) = 0$. Recalling that the probability of any formula in $\mathcal{L}$ is equivalent to the sum of complete conjunctions corresponding to its classical models as stated in (1), one obtains the following equation:

$$\begin{aligned} 0 &= (1-p) \cdot \mathcal{P}(Av) - p \cdot \mathcal{P}(A\overline{v}) \\ &= (1-p) \cdot (\sum_{\omega \in \mathrm{Mod}(Av)} \mathcal{P}(\omega)) - p \cdot (\sum_{\omega \in \mathrm{Mod}(A\overline{v})} \mathcal{P}(\omega)) \\ &= \sum_{\omega \in \mathrm{Mod}(Av)} (1-p)x_\omega - \sum_{\omega \in \mathrm{Mod}(A\overline{v})} p \cdot x_\omega. \end{aligned}$$

This is a linear equation for the variables $x_\omega$ for $\mathcal{P}$. $k^n$ linear independent equations are sufficient to determine $\mathcal{P}$ uniquely, this can be seen as a first upper bound for the size of a solution set of the learning problem. With respect to the fact that $\mathcal{P}$ can also be defined by $k^n$ assignments to the elementary worlds, this is indeed an upper bound. But recalling that in applications a dataset $\mathcal{D}$ induces the probability distribution, there will be predominantly nullconjunctions if there are many variables considered. Further assuming, that there are more efficient ways to represent the nullconjunctions, they can be handled separately and the number of variables is bounded by $O(|\mathcal{D}|)$. In fact it will be only a fraction of $|\mathcal{D}|$, since many instances $d \in \mathcal{D}$ will be represented by the same worlds. In this way a manageable bound is obtained.

### 3.2 The Problem of Minimal ME-learning

Minimal ME-Learning can be considered as a combinatorial optimization problem. In general a combinatorial optimization problem is described by a set $\mathcal{X}$ of variables with corresponding domains, constraints among the variables and an objective function that is to be optimized. Considering minimal ME-Learning, the variables $x_\phi \in \mathcal{X}$ correspond to the consistent conditionals $\phi \in (\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}}$ with respect to $\mathcal{P}$. Their domain is $\{0,1\}$, indicating if the conditional is used. The only constraint is $\mathrm{ME}(\mathcal{R}) = \mathcal{P}$ whereas $\mathcal{R} = \{\phi \in (\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}} | x_\phi = 1\}$. This constraint will have to be relaxed to $\mathrm{ME}(\mathcal{R}) \approx \mathcal{P}$ in general, since ME-inference is done by approximative approaches. The objective function to be minimized is $\sum_{x_\phi \in \mathcal{X}} x_\phi$, i.e., the number of conditionals used.

To obtain the number of variables in $\mathcal{X}$, if there are no nullconjunctions, consider the conditionals from $(\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}}$ arranged in layers. Layer $l$ includes the single-elementary conditionals with antecedence of length $l$. If there are n variables in $\mathcal{V}$, the minimum size of the antecedence is 0 ($\top$), the maximum size n-1,

hence altogether there are n layers. To each consequence literal there are $\binom{n-1}{l}$ possible combinations of antecedence variables in layer $l$. Each of these combinations may be interpreted in $k^l$ ways. So there are $\binom{n-1}{l} \cdot k^l \cdot n$ possible conditionals in layer l per consequence literal. Considering all $k \cdot n$ possible consequence literals, there are $|(\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}}| = k \cdot n \cdot \sum_{0 \leq l \leq n-1} \binom{n-1}{l} \cdot k^l = k \cdot n \cdot (k+1)^{n-1} > n \cdot k^n$ variables in $\mathcal{X}$.

As stated above, a *valid solution* of the ME-learning problem is a set $\mathcal{R}$ of conditionals, satisfying $\mathrm{ME}(\mathcal{R}) = \mathcal{P}$. Only a small proportion of the $2^{|(\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}}|}$ subsets are valid solutions, nevertheless the calculation illustrates the possible dimension of the search space of the combinatorial optimization problem. Therefore it is essential to take the nullconjunctions into account to reduce the number of possible conditionals $|(\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}}|$ to the number of consistent conditionals with respect to $\mathcal{P}$ as stated in Section 2.1.

# 4 Top-down Learning

A simple approach to overcome the problems of CondorCKD is to start with a set of most general conditionals and specialize them until $\mathcal{P}$ is sufficiently approximated. Instead of bottom-up deleting and shortening the conditionals, they are top-down added and extended. In the following a greedy top-down-approach is developed and subsequently experimentally evaluated with respect to runtime performance and knowledge base size.

## 4.1 Probabilistic Difference Measures

ME-inference is done by approximative approaches, hence in general one cannot expect equality between the original and the inferred distribution and has to define a tolerance threshold. Adequate difference measures are provided by the concept of *f-divergences* [3]. The $f$-divergence $\mathrm{D}_f(\mathcal{P}_1|\mathcal{P}_2)$ of $\mathcal{P}_1$ and $\mathcal{P}_2$ is induced by a convex function $f$, with $f(1) = 0$, as follows:

$$\mathrm{D}_f(\mathcal{P}_1|\mathcal{P}_2) = \sum_{\omega \in \Omega} \mathcal{P}_2(\omega) f(\frac{\mathcal{P}_1(\omega)}{\mathcal{P}_2(\omega)}).$$

Similar to the definition of relative entropy here $0 \cdot f(\frac{0}{0}) := 0$ and $\mathcal{P}_2(\omega) = 0$ implies $\mathcal{P}_1(\omega) = 0$ is demanded. It is easy to check that $\mathrm{D}_f = \mathrm{R}$ holds for $f(t) = t \cdot \log t$, hence the relative entropy is a special case of an $f$-Divergence. More popular measures can be derived, see [3] for an overview. The most important features they all have in common is specified by the definition of an $f$-divergence. $f(1) = 0$ assures that there is no error term $\mathcal{P}_2(\omega) f(\frac{\mathcal{P}_1(\omega)}{\mathcal{P}_2(\omega)})$ in the sum, if there is no difference between the probabilities. The convexity assures that the error term is not decreasing if the difference is increasing. The factor $\mathcal{P}_2(\omega)$ weighs the error at the point $\omega$ with respect to the probability of $\omega$. It seems reasonable that more probable worlds are also more important. Since the original distribution $\mathcal{P}$ reflects the searched probabilities it should be used as the second parameter for $\mathrm{D}_f$.

### 4.2 Decomposing the Problem

To reduce the number of variables the learning problem is decomposed into the search for a deterministic and a non-deterministic knowledge base. The following proposition states some properties of conditionals satisfyable with respect to a given probability distribution $\mathcal{P}$.

**Proposition 1.** *Let $\mathcal{P}$ be a probability distribution and let $(B|A)[x]$ be a conditional. Then the following holds.*

1. *If $\mathcal{P} \models (B|A)[x]$ then $\mathcal{P}(AB) > 0$ or $\mathcal{P}(A\overline{B}) > 0$.*
2. *$\mathcal{P} \models (B|A)[0]$ if and only if $\mathcal{P}(AB) = 0$ and $\mathcal{P}(A\overline{B}) > 0$.*
3. *$\mathcal{P} \models (B|A)[1]$ if and only if $\mathcal{P}(AB) > 0$ and $\mathcal{P}(A\overline{B}) = 0$.*
4. *If $\mathcal{P} \models (B|A)[x]$ then $x \in ]0,1[$ if and only if $\mathcal{P}(AB) > 0$ and $\mathcal{P}(A\overline{B}) > 0$.*

*Proof.* 1. If $\mathcal{P} \models (B|A)[x]$ then $0 < \mathcal{P}(A) = \mathcal{P}(A\overline{B}) + \mathcal{P}(AB)$, hence $\mathcal{P}(AB) > 0$ or $\mathcal{P}(A\overline{B}) > 0$.

2. If $\mathcal{P} \models (B|A)[0]$ then necessarily $\mathcal{P}(AB) = 0$ and $\mathcal{P}(A\overline{B}) > 0$. If conversely $\mathcal{P}(AB) = 0$ and $\mathcal{P}(A\overline{B}) > 0$ obviously $\mathcal{P} \models (B|A)[0]$ holds.

3. If $\mathcal{P} \models (B|A)[1]$ then $\mathcal{P}(A\overline{B}) = (\mathcal{P}(A\overline{B}) + \mathcal{P}(AB)) - \mathcal{P}(AB) = \mathcal{P}(A) - \mathcal{P}(B|A)\mathcal{P}(A) = 0$ and $\mathcal{P}(AB) > 0$ because of 1. If conversely $\mathcal{P}(A\overline{B}) = 0$ and $\mathcal{P}(AB) > 0$ then $\mathcal{P}(B|A) = \frac{\mathcal{P}(AB)}{\mathcal{P}(AB)+0} = 1$, hence $\mathcal{P} \models (B|A)[1]$.

4. This follows immediately from 1-3.

A probability distribution satisfying $\mathcal{P}(\omega) > 0$ for all $\omega \in \Omega$ is called *strictly positive*. An immediate consequence of the proposition is, that there can be no deterministic condionals satisfied in such a distribution, unless they are deterministic in a logical sense already, ie., $\overline{A} \vee B$ is contradictory or tautological.

Any probability distribution $\mathcal{P}$ induces a strictly positive probability distribution $\mathcal{P}^+$ by restricting the domain to the set of worlds $\Omega^+_{\mathcal{P}} := \{\omega \in \Omega | \mathcal{P}(\omega) > 0\}$ with positive probability, i.e., $\mathcal{P}^+(\omega) = \mathcal{P}(\omega)$ for all $\omega \in \Omega^+_{\mathcal{P}}$. By restricting the set of models to $\Omega^+_{\mathcal{P}}$ each formula $\mathcal{F} \in \mathcal{L}$ can be evaluated with respect to $\mathcal{P}^+$ and $\mathcal{P}(\mathcal{F}) = \sum_{\omega \in \mathrm{Mod}(\mathcal{F})} \mathcal{P}(\omega) = 0 + \sum_{\omega \in \mathrm{Mod}(\mathcal{F}) \cap \Omega^+_{\mathcal{P}}} \mathcal{P}^+(\omega) = \mathcal{P}^+(\mathcal{F})$. Hence $\mathcal{P} \models (B|A)[x]$ if and only if $\mathcal{P}^+ \models (B|A)[x]$. Since the considered datasets in the learning problem will probably contain only a fraction of the possible worlds, it seems reasonable to learn from the corresponding induced strictly positive probability distribution $\mathcal{P}^+$. In this way the number of variables might be significantly smaller, this might speed up the inference procedure for determining the solution quality as well as the learning procedure. Worlds with probability 0 are implicitly satisfied by $\mathcal{P}^+$. Hence in an optimal learning algorithm there will be no deterministic conditionals found, since they add no new information. In this way it will only determine a non-deterministic knowledge base.

To determine a deterministic knowledge base of minimal size it seems reasonable to determine the most general nullconjunctions only, since any specialization of a nullconjunction has to be a nullconjunction, too. They can be obtained by starting with all atoms and conjoining them by conjunction until their probability is 0. Let $\mathcal{N}$ denote the set of most general nullconjunctions. Then for each conjunction $N \in \mathcal{N}$ a fact $(N|\top)[0]$ can be added to the non-deterministic knowledge base for $\mathcal{P}^+$ to obtain the complete knowledge base for $\mathcal{P}$.

### 4.3 Greedy Top-down Learning

In the following a greedy top-down approach shall be considered determining a complete and compact, but not necessarily minimal knowledge base for $\mathcal{P}$. More precisely, it increases the empty conditional set successively by preferably short conditionals to a knowledge base for $\mathcal{P}$. Its search space is not built up of valid solutions of the combinatorial optimization problem from the power set $2^{(\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}}}$ but of single-elementary conditionals from $(\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}}$ that will be combined to a solution. For clear discrimination in the following the two search spaces are denoted by $\mathcal{S}_{2^{(\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}}}}$ and $\mathcal{S}_{(\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}}}$ respectively. The search space $\mathcal{S}_{(\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}}}$ can be considered as a layered graph, similar to the consideration in Section 3.2. In layer $l$ there are the valid single-elementary conditionals with antecedence of length $l$. There is an edge from conditional $(v|A)$ in layer $l$ to a conditional $(w|B)$ in layer $k$ iff $(w|B)$ is in the following layer, i.e., $l+1=k$, and they share the same consequence variable, i.e., $v=w$. That is, the successors of a conditional are obtained by adding exactly one literal, containing a variable that is not already contained in the conditional, to the antecedence.

*Example 3.* Consider three boolean variables $A, B, C$ and let $X$ denote $X = true$ for $X \in \{A, B, C\}$. Figure 3 shows the search space $\mathcal{S}_{(\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}}}$ for this example. It is artificially spanned by the conditional $(\top|\top)$. Since there are only binary variables it is sufficient to consider only the positive consequence.

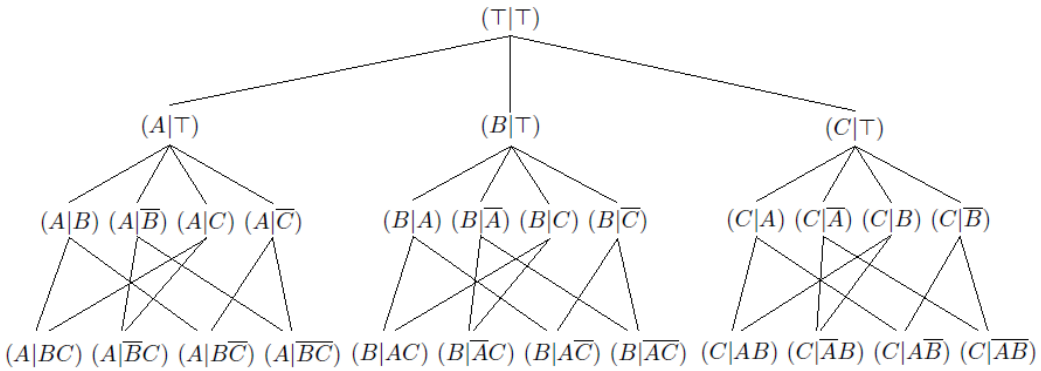


**Fig. 3.** Search space for the opening procedure

The set of currently considered conditionals is called the search space border and is denoted by $\mathcal{B}$. $\mathcal{S}_{(\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}}}$ will be searched top-down by successively expanding conditionals from the border, i.e., adding their successors to the border. Figure 4 illustrates a simple greedy algorithm.

GreedyCKD takes a probability distribution $\mathcal{P}$, an $f$-divergence $\mathrm{D}_f$ and a threshold $\epsilon$ determining the approximation accuracy as input and computes a

**Input:** probability distribution $\mathcal{P}$, $f$-divergence $\mathrm{D}_f$, threshold $\epsilon$
**Output:** conditional set $\mathcal{R}$

**BEGIN**
$\mathcal{R} \leftarrow \emptyset$, $\mathcal{B} \leftarrow \{(\top|\top)\}$, $r^* \leftarrow (\top|\top)$
$\mathcal{N} \leftarrow$ calculateNullconjunctions($\mathcal{P}$)
$\mathcal{P} \leftarrow \mathcal{P}^+$
**WHILE** $\mathcal{B} \neq \emptyset$ **AND** $\mathrm{D}_f(\mathrm{ME}(\mathcal{R}), \mathcal{P}) > \epsilon$ **DO**
$\mathcal{B} \leftarrow (\mathcal{B} \cup \mathrm{expand}(r^*, \mathcal{N})) \setminus \{r^*\}$
evaluate($\mathcal{B}$)
$r^* \leftarrow \mathrm{best}(\mathcal{B})$
$\mathcal{R} \leftarrow \mathcal{R} \cup \{r^*\}$
**END WHILE**
$\mathcal{R} \leftarrow \mathcal{R} \cup \mathrm{asFacts}(\mathcal{N})$
**RETURN** $\mathcal{R}$
**END**

**Fig. 4.** GreedyCKD

set of conditionals $\mathcal{R}$ with $\mathrm{D}_f(\mathrm{ME}(\mathcal{R}), \mathcal{P}) \leq \epsilon$. It starts with an empty conditional set and the trivial conditional $(\top|\top)$ spanning the search space $\mathcal{S}_{(\mathcal{V}|\mathcal{L})^{\wedge}_{\mathcal{P}}}$, i.e., $(\top|\top)$ is expanded by the conditionals $(v|\top)$ for all $v \in \mathrm{dom}(V)$ and all $V \in \mathcal{V}$. $r^*$ represents the currently best conditional. The most general nullconjunctions are calculated separately and $\mathcal{P}$ is replaced by $\mathcal{P}^+$. In this way it is in particular possible to guarantee well-definedness for $\mathrm{D}_f$, since it is assured that a nullconjunction in the original distribution is a nullconjunction in the inferred distribution, too. The main loop is executed until the search space border is empty or the correctness criteria is reached. Therein the currently best conditional is expanded and removed from the search space border. The expansion procedure has to take the nullconjunctions into account to pass only consistent and non-deterministic conditionals. The conditionals in the search space border are then evaluated, e.g., with respect to the obtained approximation gain, if adding the current conditional to $\mathcal{R}$. Subsequently the (locally) best conditional is selected in a greedy manner.

Assuming the basic conditionals represent the distribution exactly, the algorithm will always terminate satisfying $\mathrm{D}_f(\mathrm{ME}(\mathcal{R}), \mathcal{P}^+) \leq \epsilon$. Since the nullworlds are covered in the last step by adding the nullconjunctions as facts, it finally holds $\mathrm{D}_f(\mathrm{ME}(\mathcal{R}), \mathcal{P}) \leq \epsilon$.

**Naive Top-down Implementation (TDCKD):** A naive version has been implemented first. Conditionals are expanded by all successors, i.e., by extending the antecedence with values of variables not already contained in the conditional. The search space border is evaluated by calculating the approximate gain ([1], [4]) for every border conditional $b \in \mathcal{B}$. Basically it approximates $\mathrm{D}_f(\mathrm{ME}(\mathcal{R} \cup$

$\{b\}), \mathcal{P})$ by only adapting the factor for the new conditional $b$ in the inference step. The conditional obtaining the minimum value is selected and added to $\mathcal{R}$.

So far only relative entropy has been used as a difference measure. If there are long condionals in the set inducing $\mathcal{P}$, the algorithm tends to find many short rules that are specialized by longer rules subsequently and therefore might be redundant. Nevertheless, they cannot be removed readily, unless every direct successor has been added to the knowledge base. While CondorCKD tends to represent few short conditionals by many long conditionals, GreedyCKD represents few long conditionals by many short conditionals.

**Frequent Pattern Implementation (FPCKD):** Since the simple implementation has to add many potentially redundant short conditionals before obtaining interesting long conditionals, it seems reasonable to add a set of promising conditionals to the search space border initially. A machine learning area with a strong structural similarity to ME-learning is the area of association rule learning [6]. There the search for interesting rules is often decomposed by looking for frequent patterns first and subsequently generating rules from these. Ideas from [13] and [12] have been adapted to detect frequent patterns, i.e., atoms appearing frequently together, and generate promising single-elementary conditionals from them.

### 4.4 Experimental Results

To compare the implementations TDCKD and FPCKD to CondorCKD, modifications of the US Census Data from the UCI Machine Learning Repository have been used [5]. Since all algorithms transform the dataset into a probability distribution their runtime is almost independent of the number of instances, but strongly dependent on the number of possible worlds induced by the variables. Hence different subsets of the contained variables have been considered. TDCKD and FPCKD had to approximate the original distribution to a threshold of $10^{-4}$. Since CondorCKD contains no backtracking so far, the learned knowledge base does not necessarily represent the original distribution again, but might represent it much better as well. To each algorithm one 2,000 MHz CPU and 256 MB of main memory were assigned.

Figure 4.4 compares the runtime performance for the three implementations. In the first picture the test started with 64 possible worlds (6 binary variables) and was repeated with 128 (additional binary variable), 384 (additional ternary variable) and 1152 (additional ternary variable) possible worlds. Using CondorCKD there is a tradeoff between runtime and approximation quality. If many dependencies are found, there will be many shortenings necessary. This results in a smaller knowledge base but increasing runtime. Conversely, if only few dependencies are found, the algorithm can be much faster, but returns a huge knowledge base of unshortened and highly redundant conditionals. It has been configured as a compromise between a short knowledge base and good performance, but better configurations might be possible. Using different configurations it has been clearly outperformed at 384 worlds. The test set of the second

picture started with two ternary and two quarternary variables (144 worlds), subsequently binary variables had been added. The runtime of CondorCKD exceeds the diagram for the first set already.

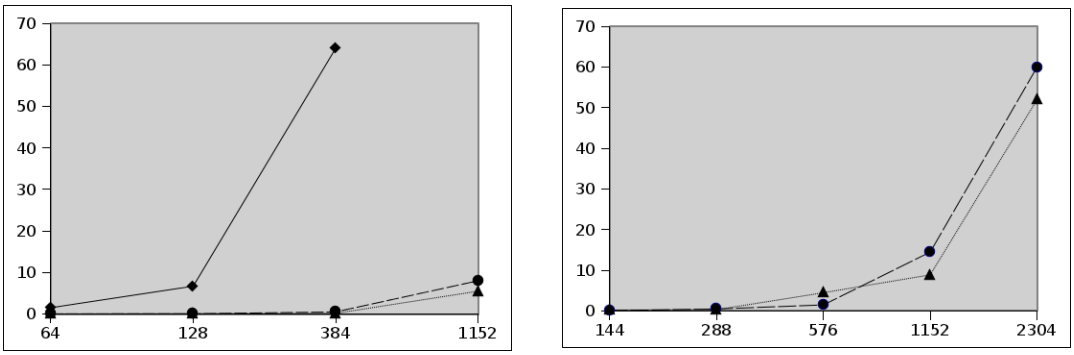


**Fig. 5.** Runtime in minutes for CondorCKD (continuous), TDCKD (dashed) and FPCKD (dotted).

In figure 4.4 the size of the learned knowledge base is illustrated. The continuous line represents the upper bound as described in section 3.1. Since CondorCKD is not aimed at minimal learning it determined knowledge bases above this bound and hence is not incorporated in the comparison. On the left the number of possible worlds is compared to the number of learned conditionals describing the original distribution. On the right, inspired by information theory, the possible worlds and the learned conditionals are evaluated with respect to their description length. Let the description length of an atom $(V = v)$ be the number of bits needed to encode the value of the considered variable, i.e., $\lceil \log \mathrm{dom}(V) \rceil$. The description length of a conjunction is the sum of the description lengths of the contained atoms and the description length of a conjunctive conditional is the sum of the description lengths of the antecedence and the consequence. Using the language from example 1 the description length of the world $(A = new) \wedge (P = high) \wedge (F = petrol) \wedge (C = high)$ is $1+2+1+2 = 6$ and the description length of the conditional $(C = high|P = high)[1]$ is $2+2 = 4$. As the figures show the knowledge base representation becomes much more rewarding, if the number of possible worlds is growing.

Both top-down approaches perform similar with respect to the runtime as well as with respect to the knowledge base size. For an increasing number of variables FPCKD becomes slightly faster, since the frequent pattern structure can be used to determine satisfying and violating worlds in the beginning more efficient. For a small number of variables the advantage is outweighed by the effort to determine frequent patterns.

It is also worth mentioning that TDCKD finds the smaller knowledge base in most cases, even though it does not make use of any information but the current information gain of the currently considered conditionals. In fact FPCKD determines larger databases if the minimum support is lower and thus the initial

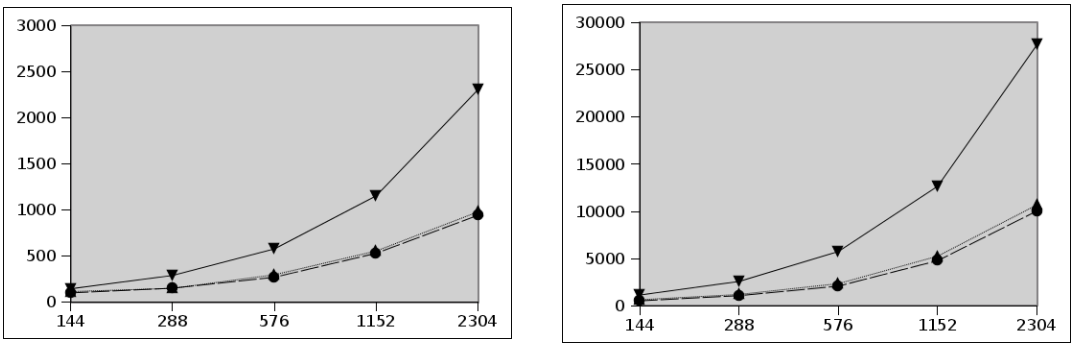


**Fig. 6.** Upper bound (continuous) for the knowledge base size and results for TDCKD (dashed) and FPCKD (dotted) with respect to required conditionals (left) and description length (right).

search space border is larger. The operating sequence of TDCKD can be intuitively described as consecutively adding general assumptions in form of facts and delimiting them with respect to further information subsequently. This approach seems to work well for ME-learning, at least for problems of moderate size.

## 5 Discussion and Further Work

In this paper a top-down ME-learning approach has been developed focused on minimal learning rather than on inverting ME-inference in a rigorous sense. The search for a minimal knowledge base is divided into the search for a deterministic and a probabilistic part to speed up the calculation steps. The two implementations TDCKD and FPCKD perform well for problems of moderate size.

One performance weakness is the search for nullconjunctions to determine the deterministic part of the knowledge base. Currently it follows a brute force approach that should be replaced by a more sophisticated method. A heuristical tree structure approach used in CondorCKD [9] might be more efficient.

Recalling the upper bound for a knowledge base from section 3.1 the search for the probabilistic part can be executed in $O(|\mathcal{D}|)$ iterations. If the approximation is not sufficient at this point, one can as well return the worlds in $\mathcal{D}$ as facts as a solution. The computation-costly steps in each iteration are firstly the calculation of the satisfying and violating worlds for the conditionals added to the search space border and the evaluation of the conditionals. Both steps can be performed in time $O(|\mathcal{D}|)$ and can be well parallelized. Furthermore, after each iteration the currently represented probability distribution has to be derived, this will become challenging for large problems.

To reduce the number of expensive inference steps it seems reasonable to add several promising conditionals in each step. This will result in a bigger knowledge base very likely. To further minimize it, subsequently common metaheuristics

like simulated annealing could be used, see [2] for an overview. The mentioned dependencies between numerical and conditional structure as described in [8] already used in CondorCKD could be helpful to detect redundancies.

Even though a significant performance gain is obtained, the steep raising indicates that this approach is not capable of handling datatypes containing more than some thousand possible worlds, i.e., about 15 binary or some multi-valued variables. Approximative bayesian network learning approaches scale much better in this order of magnitude. Since these approaches are aimed at most probable rather than sufficiently exact approximations of the data, this performance will probably never be obtained. However, it is certainly possible to improve performance by increasing the tolerance threshold or using a time limit. Roughly speaking the algorithms would determine the most informative conditionals in this way. With respect to interpretability the frequent pattern approach seems more convenient for this purpose.

## References


1. Berger, A.L., Pietra, V.J.D., Pietra, S.A.D.: A maximum entropy approach to natural language processing. Comput. Linguist. 22, 39–71 (March 1996)
2. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Comput. Surv. 35, 268–308 (September 2003)
3. Csiszár, I., Shields, P.C.: Information theory and statistics: a tutorial. Commun. Inf. Theory 1, 417–528 (December 2004)
4. Dehaspe, L.: Maximum entropy modeling with clausal constraints. In: Proceedings of the 7th International Workshop on Inductive Logic Programming. pp. 109–124. Springer-Verlag, London, UK (1997)
5. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml
6. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. Data Min. Knowl. Discov. 15, 55–86 (August 2007)
7. Jaynes, E.: Papers on Probability, Statistics and Statistical Physics. D. Reidel Publishing Company, Dordrecht, Holland (1983)
8. Kern-Isberner, G.: Conditionals in nonmonotonic reasoning and belief revision. Springer, Lecture Notes in Artificial Intelligence LNAI 2087 (2001)
9. Kern-Isberner, G., Fisseler, J.: Knowledge discovery by reversing inductive knowledge representation. In: Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning, KR-2004. pp. 34–44. AAAI Press (2004)
10. Kindermann, R.: Markov Random Fields and Their Applications. Amer Mathematical Society, http://www.worldcat.org/isbn/0821850016
11. Neapolitan, R.E.: Learning Bayesian Networks. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (2003)
12. Zaki, M.J.: Generating non-redundant association rules. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 34–43. KDD '00, ACM, New York, NY, USA (2000)
13. Zaki, M.J.: Scalable algorithms for association mining. IEEE Trans. on Knowl. and Data Eng. 12, 372–390 (May 2000)

**Verzeichnis der zuletzt erschienenen Informatik-Berichte**

[344] Behr, T., Güting, R. H.:
User Defined Topological Predicates in Database Systems

[345] vor der Brück, T.; Helbig, H.; Leveling, J.:
The Readability Checker Delite Technical Report

[346] vor der Brück, T.:
Application of Machine Learning Algorithms for Automatic Knowledge Acquisition and Readability Analysis Technical Report

[347] Fechner, B.:
Dynamische Fehlererkennungs- und –behebungsmechanismen für verlässliche Mikroprozessoren

[348] Brattka, V., Dillhage, R., Grubba, T., Klutsch, A.:
CCA 2008 - Fifth International Conference on Computability and Complexity in Analysis

[349] Osterloh, A.:
A Lower Bound for Oblivious Dimensional Routing

[350] Osterloh, A., Keller, J.:
Das GCA-Modell im Vergleich zum PRAM-Modell

[351] Fechner, B.:
GPUs for Dependability

[352] Güting, R. H., Behr, T., Xu, J.:
Efficient *k*-Nearest Neighbor Search on Moving Object Trajectories

[353] Bauer, A., Dillhage, R., Hertling, P., Ko K.I., Rettinger, R.:
CCA 2009 Sixth International Conference on Computability and Complexity in Analysis

[354] Beierle, C., Kern-Isberner, G.
Relational Approaches to Knowledge Representation and Learning

[355] Sakr, M.A., Güting, R.H.
Spatiotemporal Pattern Queries

[356] Güting, R. H., Behr, T., Düntgen, C.:
SECONDO: A Platform for Moving Objects Database Research and for Publishing and Integrating Research Implementations

[357] Düntgen, C., Behr, T., Güting, R.H.:
Assessing Representations for Moving Object Histories

[358] Sakr, M.A., Güting, R.H.
Group Spatiotemporal Pattern Queries

[359] Hartrumpf, S., Helbig, H., vor der Brück, T. , Eichhorn, C.
SemDupl: Semantic Based Duplicate Identification

[360] Xu, J., Güting, R.H.
A Generic Data Model for Moving Objects