Using Maximum Entropy in a Defeasible Logic with Probabilistic Semantics

James Cussens[1] and Anthony Hunter[2]

[1] Centre for Logic and Probability in IT, King's College
Strand, London, WC2R 2LS, UK
email: j.cussens@elm.cc.kcl.ac.uk

[2] Department of Computing, Imperial College
180 Queen's Gate, London, SW7 2BZ, UK
email: abh@doc.ic.ac.uk

Abstract. In this paper we make defeasible inferences from conditional probabilities using the Principle of Total Evidence. This gives a logic that is a simple extension of the axiomatization of probabilistic logic as defined by Halpern's AX_1. For our consequence relation, the reasoning is further justified by an assumption of the typicality of individuals mentioned in the data. For databases which do not determine a unique probability distribution, we select by default the distribution with Maximum Entropy. We situate this logic in the context of preferred models semantics.

1 Introduction

Traditionally there has been a dichotomy between the probabilistic and logical views on uncertainty reasoning in AI. However, there does seem to be an intuitive overlap of the two views. Whilst there are logics capturing aspects of probabilistic reasoning ([1, 3, 6, 9]), the formal relationship of defeasible logics with probability theory remains an interesting research topic. In particular, using probability theory gives us an opportunity to clarify aspects of non-monotonic logics.

Here we show how using established principles from probability theory, together with the axioms of probability, can provide a useful semantic foundation for a defeasible logic. For the logic we extend the approach of [6] to allow non-monotonic reasoning from a database of conditional probability statements and ground formulae. We justify such non-monotonic inferences by the Principle of Total Evidence. We use this when we have two or more conditional probabilities all of whose conditions are satisfied. From these, the principle selects the conditional probability with the most specific condition. The idea here is that when assessing the probability of a given event, we should calculate its probability value conditional on all the available evidence. It is clear that the probability-theoretic Principle of Total Evidence is analogous to the notion of specificity as used in defeasible logic.

If there is no conditional probability statement that matches the available evidence, then we can estimate the required probability statement by invoking the Principle of Maximum Entropy. This gives the least biased estimate given the conditional probability statements already in the database ([8, 2, 10]).

In the following we provide an overview of the language, form of database and consequence relation for the defeasible logic; together with an outline of the formal semantics. We also discuss how we use Maximum Entropy to support reasoning in this logic.

2 A Representation for Defeasible Reasoning

We use Halpern's language $L_1^=(\Phi)$ together with the axiomatization AX_1 for $L_1^=(\Phi)$ which is correct for finite domains. Essentially this extends first-order classical logic to give a logic of probability. We use $\vdash$ to denote the consequence relation for AX_1. We reason with two kinds of information: We have conditional probability statements that give statistical information, and we also have a scenario—a set of ground formulae.

A probabilistic database is a pair (Δ, Σ), where Δ is a finite set of conditional probability statements in the language $L_1^=(\Phi)$ and Σ, the scenario, is a finite set of closed object formulae in $L_1^=(\Phi)$. The conditional probability terms are restricted to the form, $w_x(\alpha(x)|\beta(x))$, where both α and β are formulae with x free, s.t. x is a tuple of variables, β is classically consistent and there are no further variables or function symbols in either α or β. These are called *completely bound* probability terms. The conditional probability statements are restricted to the form $w_x(\alpha(x)|\beta(x)) = \zeta$, where $w_x(\alpha(x)|\beta(x))$ is a completely bound probability term, and ζ is real. We abbreviate terms of the form $w_x(\alpha(x)|\beta(x) \vee \neg\beta(x))$ to $w_x(\alpha(x))$.

A probabilistic database (Δ, Σ) is complete w.r.t./ Φ if, for every completely bound probability term $w_x(\alpha(x)|\beta(x))$ in $L_1^=(\Phi)$, there is a ζ s.t. $\Delta \vdash w_x(\alpha(x)|\beta(x)) = \zeta$ holds. However, if a database is not complete then we can complete it using Maximum Entropy. We discuss this below.

3 A Consequence Relation for Defeasible Reasoning

Given a complete probabilistic database (Δ, Σ), we wish to ascertain a probability value for ground literals α if $\alpha \notin \Sigma$. If we use the Principle of Total Evidence, then there is only one conditional probability statement in the database that can be used. If the value of this conditional probability is greater than a certain threshold, say θ, then we defeasibly infer α. Note, we must have θ at least $1/2$ for consistency. If the value is less than $1 - \theta$, then we defeasibly infer $\neg\alpha$. However, if the value is in $[1 - \theta, \theta]$, then we infer neither α nor $\neg\alpha$.

We formalize an inference from a database as follows, where θ is the probabilistic threshold value, and the relation $\mid\!\sim$ denotes non-monotonic consequence. Let σ be a function that assigns every constant in Φ a distinct variable. Given a formula α, let α^σ be the result of replacing every constant c that appears in α by the variable $\sigma(c)$. Let $\sigma(\alpha)$ be the set of variables appearing in α^σ. The conjunct $\bigwedge(\Sigma)$ is formed from all the elements in Σ, and $\vdash$ is the consequence relation defined by Halpern for $L_1^=(\Phi)$:

$$(\Delta, \Sigma) \mid\!\sim \alpha \text{ iff } \Delta \vdash w_{\sigma(\Sigma)}(\alpha^\sigma|(\bigwedge(\Sigma))^\sigma) > \theta \tag{1}$$

The non-monotonic consequence relation captures the notion of reasoning from a preferred conditional probability for a query α, where it is preferred on the basis of the Principle of Total Evidence. The definition ensures the following: (1) the chosen conditional probability is greater than θ; and (2) the function σ makes the head of the conditional probability equivalent to the query and the antecedent equivalent to the scenario.

4 An Example of Defeasible Reasoning

We take the bird example, where the database (Δ, Σ) is defined as follows, and $\Phi = \{\text{bird}, \text{fly}, \text{penguin}, \text{polly}, \text{tweety}\}$

$$\begin{aligned}
\Delta = \{ & w_x(\text{bird}(x)) = p_1 \\
& w_x(\text{fly}(x)) = p_2 \\
& w_x(\text{penguin}(x)) = p_3 \\
& w_x(\text{fly}(x)|\text{bird}(x)) = p_4 \\
& w_x(\text{bird}(x)|\text{penguin}(x)) = p_5 \\
& w_x(\text{penguin}(x)|\text{fly}(x)) = p_6 \\
& w_x(\text{fly}(x)|\text{bird}(x) \wedge \text{penguin}(x)) = p_7 \}
\end{aligned} \tag{2}$$

$$\Sigma = \{\text{bird}(\text{tweety}), \text{bird}(\text{polly})\} \tag{3}$$

For query fly(tweety), we have: $\Delta \vdash w_x(\text{fly}(x)|\text{bird}(x)) = p_4$. Now, if $p_4 > \theta$ holds, then by AX_1 and (1), we have $(\Delta, \Sigma) \mid\!\sim \text{fly}(\text{tweety})$.

We extend the scenario to the following,

$$\Sigma' = \{\text{bird}(\text{tweety}), \text{bird}(\text{polly}), \text{penguin}(\text{tweety})\} \tag{4}$$

For query $\neg$fly(tweety) from the database (Δ, Σ'), we have the following:

$$\Delta \vdash w_x(\neg\text{fly}(x)|\text{bird}(x) \wedge \text{penguin}(x)) = 1 - p_7 \tag{5}$$

Now, if $(1 - p_7) > \theta$ holds, then by AX_1 and (1), we have $(\Delta, \Sigma') \mid\!\sim \neg\text{fly}(\text{tweety})$.

5 A Probabilistic Semantics for the Logic

To provide a semantics for the defeasible logic, we extend the type-1 semantics ([6]) which is correct with respect to the AX_1 axiomatization for finite domains—for this paper we assume all domains are finite. A type-1 probability structure is a tuple (D, π, F, μ) where D is a domain, π assigns to the predicate and function symbols in the language predicates and functions of the right arity over D (so that (D, π) is just a standard first-order structure). F is a finite algebra generated from the set of all sets definable in $L_1^=(\Phi)$, and μ is a probability function on F. We define the atoms of F, denoted $A(F)$, as the minimal non-empty sets of F. μ is determined by the values it takes on $A(F)$.

A valuation v, is a function mapping each object variable into an element of D. Given a type-1 probability structure M and valuation v, we proceed by induction to associate with every object (respectively field) term t, an element $[t]_{(M,v)}$ of D (respectively $\mathbb{R}$), and with every formula ϕ a truth value, writing $(M, v) \models \phi$ if the value true is associated with ϕ by (M,v). We write $M \models \phi$ if $(M,v) \models \phi$ for all valuations v. We also define $D_\phi = \{d \in D : (M, v[x/d]) \models \phi\}$. Finally for every probability term $w_x(\phi)$, $[w_x(\phi)]_{(M,v)} = \mu^n(D_\phi)$ where n is the number of free variables in ϕ, and μ^n is the product measure. We also add the following to the definition of satisfaction: $(M,v) \models (\Delta, \Sigma)$ iff $\forall \phi \in \Delta \cup \Sigma : (M,v) \models \phi$. For this definition, if $(M,v) \models (\Delta,\Sigma)$ for some valuation v, then $(M,v) \models (\Delta,\Sigma)$ for all valuations v, since (Δ,Σ) contains no free variables. So we can replace $(M,v) \models (\Delta,\Sigma)$ by $M \models (\Delta,\Sigma)$.

Using this semantics it is straightforward to define a corresponding notion of semantics for the defeasible logic consequence relation, and show correctness results. For this semantics we take the set of models for (Δ,Σ) and select a subset of models, denoted $[[\Delta,\Sigma]]$. This subset of models is the set of preferred models for the database, such that $M \in [[\Delta,\Sigma]]$ if, whenever $M \models (\Delta,\Sigma)$ and $M \models w_{\sigma(\Sigma)}(\alpha^\sigma | (\bigwedge(\Sigma))^\sigma) > \theta$, then $M \models \alpha$. Using preferred models builds on the non-monotonic logics framework initially proposed by [12]. We define preferred satisfaction as follows:

$$M \mathrel{|\!\!\approx_\Sigma} \alpha \text{ iff } M \models w_{\sigma(\Sigma)}(\alpha^\sigma | (\bigwedge(\Sigma))^\sigma) > \theta \tag{6}$$

Using this, we define preferred entailment, denoted $\mathrel{|\!\!\approx}$, as follows:

$$\forall M (M \models (\Delta,\Sigma) \Rightarrow M \mathrel{|\!\!\approx_\Sigma} \alpha) \text{ iff } (\Delta,\Sigma) \mathrel{|\!\!\approx} \alpha \tag{7}$$

We now consider properties of this form of non-monotonic reasoning. For a database (Δ,Σ), we use the notation $(\Delta,\Sigma) \vdash \alpha$ to represent $(\Delta \cup \Sigma) \vdash \alpha$.

Theorem 1. *For the consequence relation $\vdash$ for $L_1^=(\Phi)$, the entailment relation $\models$ for $L_1^=(\Phi)$, and a database (Δ,Σ), where $(\Delta \cup \Sigma) \subset L_1^=(\Phi)$, and $\alpha \in L_1^=(\Phi)$, the equivalence $(\Delta,\Sigma) \vdash \alpha$ iff $(\Delta,\Sigma) \models \alpha$ holds, since we assume finite domains.*

Proof. See [6]. □

Lemma 2. *Using the definition of preferred entailment, the following equivalence holds: $(\Delta,\Sigma) \mathrel{|\!\!\approx} \alpha$ iff $(\Delta,\Sigma) \models w_{\sigma(\Sigma)}(\alpha^\sigma | (\bigwedge(\Sigma))^\sigma) > \theta$.*

Proof. Assume $(\Delta,\Sigma) \mathrel{|\!\!\approx} \alpha$ holds, then choose an arbitrary M such that $M \models (\Delta,\Sigma)$. From (7), we have $M \mathrel{|\!\!\approx_\Sigma} \alpha$, and hence from (6), $M \models w_{\sigma(\Sigma)}(\alpha^\sigma | (\bigwedge(\Sigma))^\sigma) > \theta$. Since M was arbitrary this shows $(\Delta,\Sigma) \models w_{\sigma(\Sigma)}(\alpha^\sigma | (\bigwedge(\Sigma))^\sigma) > \theta$. So $(\Delta,\Sigma) \mathrel{|\!\!\approx} \alpha$ implies $(\Delta,\Sigma) \models w_{\sigma(\Sigma)}(\alpha^\sigma | (\bigwedge(\Sigma))^\sigma) > \theta$. Now assume $(\Delta,\Sigma) \mathrel{|\!\!\not\approx} \alpha$. ¿From (7) there exists an M such that $M \models (\Delta,\Sigma)$, but $M \mathrel{|\!\!\not\approx_\Sigma} \alpha$. ¿From (6), this implies $M \not\models w_{\sigma(\Sigma)}(\alpha^\sigma | (\bigwedge(\Sigma))^\sigma) > \theta$. Hence $(\Delta,\Sigma) \not\models w_{\sigma(\Sigma)}(\alpha^\sigma | (\bigwedge(\Sigma))^\sigma) > \theta$. So $(\Delta,\Sigma) \models w_{\sigma(\Sigma)}(\alpha^\sigma | (\bigwedge(\Sigma))^\sigma) > \theta$ implies $(\Delta,\Sigma) \mathrel{|\!\!\approx} \alpha$ and the result follows. □

Theorem 3. *For any probabilistic database (Δ,Σ), the equivalence $(\Delta,\Sigma) \mathrel{|\!\!\approx} \alpha$ iff $(\Delta,\Sigma) \mathrel{|\!\!\sim} \alpha$ holds, where α is a ground literal.*

Proof. By (1) and Lemma 2, the result obtains by showing the equivalence $(\Delta,\Sigma) \models w_{\sigma(\Sigma)}(\alpha^\sigma | (\bigwedge(\Sigma))^\sigma) > \theta$ iff $(\Delta,\Sigma) \vdash w_{\sigma(\Sigma)}(\alpha^\sigma | (\bigwedge(\Sigma))^\sigma) > \theta$. This equivalence follows directly from Theorem 1. □

6 Properties of the Logic

It is of interest to draw an analogy with the Closed World Assumption (CWA), where extra assumptions can be drawn from the database under certain conditions. However, since our non-monotonic consequence relation is based on probability theory, we avoid some of the problems of the CWA. For pathological examples, such as the following database $\Gamma = \{\neg\alpha \rightarrow \alpha\}$, the equivalent formula in $L_1^=(\Phi)$, $w_x(\alpha(x)|\neg\alpha(x)) > \theta$ does not hold since $w_x(\alpha(x)|\neg\alpha(x)) = 0$ follows directly from probability theory.

Another interesting difference between our non-monotonic consequence relation and that of CWA is with regard to the following kind of database: For $\Gamma' = \{\neg\beta \rightarrow \alpha\}$, both $\neg\alpha$ and $\neg\beta$ follow by CWA, and the pair of inferences are inconsistent with Γ. In our approach, for a consistent database, the non-monotonic consequence relation does not allow inconsistent inferences. Indeed, it is straightforward to show that if (Δ, Σ) is satisfiable then $[[\Delta, \Sigma]]$ is non-empty, since there is no consistent (Δ, Σ) and α such that $(\Delta, \Sigma) \approx \alpha$ and $(\Delta, \Sigma) \approx \neg\alpha$.

In comparison with Gabbay's axiomatization of the consequence relation [4], cut holds if no conditional probability values deducible from Δ lie in the interval $(\theta^2, \theta]$. Similarly, cautious monotonicity holds if no conditional probability values deducible from Δ lie in the set $(\theta(1-\theta)^2, 1-\theta]$. However, in general, neither cut nor cautious monotonicity hold. Furthermore, in general, reflexivity holds and monotonicity fails.

Theorem 4. *For the consequence relation $\mid\sim$, with query α and database (Δ, Σ), the following property does not hold: $(\Delta, \Sigma) \mid\sim \alpha$ and $(\Delta, \Sigma) \mid\sim \beta$ implies $(\Delta \cup \{\alpha\}) \mid\sim \beta$.*

Proof. Suppose we have that $\Delta = \{w_x(\text{mammal}(x)) = 0.55, w_x(\text{egg-layer}(x)) = 0.55, w_x(\text{mammal}(x)|\text{egg-layer}(x)) = 0.2\}$ and θ is set to its lowest value: 0.5. We have $(\Delta, \emptyset) \mid\sim \text{mammal(agatha)}$ and $(\Delta, \emptyset) \mid\sim \text{egg-layer(agatha)}$ but not $(\Delta, \text{egg-layer(agatha)}) \mid\sim \text{mammal(agatha)})$. □

Theorem 5. *For the consequence relation $\mid\sim$, with query α and database (Δ, Σ), the following property does not hold: $(\Delta, \Sigma) \mid\sim \beta$ and $(\Delta, \Sigma \cup \{\beta\}) \mid\sim \alpha$ implies $(\Delta, \Sigma) \mid\sim \alpha$.*

Proof. Suppose that we have $\Delta = \{w_x(\text{bird}(x)) = 0.7, w_x(\text{flies}(x)) = 0.6, w_x(\text{flies}(x)|\text{bird}(x)) = 0.9\}$ and θ is set to 0.65, then we have $(\Delta, \emptyset) \mid\sim \text{bird(tweety)}$ and $(\Delta, \text{bird(tweety)}) \mid\sim \text{flies(tweety)}$, but not $(\Delta, \emptyset) \mid\sim \text{flies(tweety)}$. □

Theorem 6. *For the consequence relation $\mid\sim$, with query α and database (Δ, Σ), the following properties do hold:*

(Left logical equivalence)

$$\frac{\vdash \alpha \equiv \beta; (\Delta, \Sigma \cup \{\alpha\}) \mid\sim \gamma}{(\Delta, \Sigma \cup \{\beta\}) \mid\sim \gamma} \tag{8}$$

(Conjunctive Sufficiency)

$$\frac{(\Delta, \Sigma) \mid\sim \alpha \wedge \beta}{(\Delta, \Sigma \cup \{\alpha\}) \mid\sim \beta} \tag{9}$$

(Weak Conditionalization)

$$\frac{(\Delta, \Sigma \cup \{\alpha\}) \mathrel{|\!\sim} \beta}{(\Delta, \Sigma) \mathrel{|\!\sim} \alpha \rightarrow \beta} \tag{10}$$

(Right Weakening)

$$\frac{\Sigma \vdash \alpha \rightarrow \beta; (\Delta, \Sigma) \mathrel{|\!\sim} \alpha}{(\Delta, \Sigma) \mathrel{|\!\sim} \beta} \tag{11}$$

(Reasoning by Cases)

$$\frac{(\Delta, \Sigma \cup \{\alpha\}) \mathrel{|\!\sim} \beta; (\Delta, \Sigma \cup \{\neg\alpha\}) \mathrel{|\!\sim} \beta}{(\Delta, \Sigma) \mathrel{|\!\sim} \beta} \tag{12}$$

(Correlative Monotonicity)

$$\frac{(\Delta, \Sigma) \mathrel{|\!\sim} \beta; (\Delta, \Sigma \cup \{\beta\}) \mathrel{|\!\sim} \alpha; (\Delta, \Sigma) \not\mathrel{|\!\sim} \alpha}{(\Delta, \Sigma \cup \{\alpha\}) \mathrel{|\!\sim} \beta} \tag{13}$$

(Correlative Cut)

$$\frac{(\Delta, \Sigma) \mathrel{|\!\sim} \alpha; (\Delta, \Sigma \cup \{\alpha\}) \mathrel{|\!\sim} \beta; (\Delta, \Sigma \cup \{\beta\}) \not\mathrel{|\!\sim} \alpha}{(\Delta, \Sigma) \mathrel{|\!\sim} \beta} \tag{14}$$

(Cautious $\vee$ Introduction)

$$\frac{(\Delta, \Sigma \cup \{\alpha\}) \mathrel{|\!\sim} \gamma; (\Delta, \Sigma \cup \{\beta\}) \mathrel{|\!\sim} \gamma; \Sigma \vdash \neg(\alpha \wedge \beta \wedge \gamma)}{(\Delta, \Sigma \cup \{\alpha \vee \beta\}) \mathrel{|\!\sim} \gamma} \tag{15}$$

Proof. All of the above follow directly from the probability calculus. □

7 Completing a Probabilistic Database by Using Maximum Entropy

We can increase the applicability of this defeasible logic by providing ways of constructing a complete database from an incomplete database. A complete database fully determines a probability distribution μ over F, whereas a incomplete database only gives a set of constraints on possible probability distributions over F. Therefore to complete a database, we need to consider how to select one appropriate probability distribution from the possible distributions allowed by the incomplete database.

Here, we consider using Maximum Entropy to complete a database. This provides a unique complete database Δ^* that extends the incomplete database Δ. Essentially entropy is an inverse measure of the information contained in a probability distribution. Any probability distribution μ has an associated entropy $H(\mu)$ defined as follows, where ln equals $\log_e$.

$$H(\mu) = Q \sum_{B \in A(F)} \mu(B) \ln \mu(B) \tag{16}$$

Choosing the distribution with the Maximum Entropy hence corresponds to completing a database in an unbiased way ([8, 2, 10]). For example, consider the database $\Delta = \{w_x(\text{fly}(x)) = p_2, w_x(\text{bird}(x)) = p_3\}$ which is incomplete with respect to $\Phi = \{\text{fly}, \text{bird}\}$. To complete the database, we need to add the following conditional probability statement $w_x(\text{fly}(x)|\text{bird}(x)) = p_1$. Using Maximum Entropy, we calculate (below) that the value of p_1 is equal to the value of p_2. In this simple example, Maximum Entropy allows us to assume that the two predicates are independent.

Now consider the database $\Delta' = \{w_x(\text{fly}(x)|\text{bird}(x)) = p_1, w_x(\text{fly}(x)) = p_2, w_x(\text{bird}(x)) = p_3\}$ which is complete with respect to $\Phi = \{\text{fly}, \text{bird}\}$. Suppose we extend Φ' to $\{\text{fly}, \text{bird}, \text{sparrow}\}$, then Δ' is incomplete with respect to Φ'. Therefore we need to generate a database that include statements such as $w_x(\text{fly}(x)|\text{bird}(x) \wedge \text{sparrow}(x)) = p_4$. Using Maximum Entropy again, we can calculate that $p_1 = p_4$. In this situation, we can also see that Maximum Entropy is introducing an assumption that new properties are irrelevant to the probability distribution unless otherwise stated.

Viewing the use of Maximum Entropy as capturing a notion of relevance in probabilistic data follows the approach discussed by [11]. We can also view the use of Maximum Entropy as an inheritance principle as discussed by [5].

For example, for the database $\Delta = \{w_x(\text{fly}(x)) = p_2, w_x(\text{bird}(x)) = p_3\}$ which is incomplete with respect to $\Phi = \{\text{fly}, \text{bird}\}$, we find the value of p_1 for $w_x(\text{fly}(x)|\text{bird}(x)) = p_1$, by the use of Lagrange multipliers. The formulae in Δ provide linear constraints on any distribution satisfying Δ. For each constraint we define a function on $A(F)$, denoted f_0, f_1, and f_2 as follows:

$$f_0(A(F)) = \sum_{B \in A(F)} w_x(B(x)) - 1 = 0 \tag{17}$$

This constraint is that all the probabilities add to 1. We also have,

$$f_1(A(F)) = w_x(\text{bird}(x) \wedge \text{fly}(x)) + w_x(\neg\text{bird}(x) \wedge \text{fly}(x)) - p_2 = 0 \tag{18}$$

$$f_2(A(F)) = w_x(\text{bird}(x) \wedge \text{fly}(x)) + w_x(\text{bird}(x) \wedge \neg\text{fly}(x)) - p_3 = 0 \tag{19}$$

Entropy is maximized, subject to the above constraints, by the following scheme. For any B in $A(F)$

$$\frac{\partial H}{\partial(w_x(B(x)))} + \lambda\frac{\partial f_0}{\partial(w_x(B(x)))} + \lambda_1\frac{\partial f_1}{\partial(w_x(B(x)))} + \lambda_2\frac{\partial f_2}{\partial(w_x(B(x)))} = 0 \tag{20}$$

For each $B \in A(F)$ we differentiate to give the following:

$$-(\ln(w_x(\text{bird}(x) \wedge \text{fly}(x)) + 1) + \lambda + \lambda_1 + \lambda_2 = 0 \tag{21}$$

$$-(\ln(w_x(\text{bird}(x) \wedge \neg\text{fly}(x)) + 1) + \lambda + \lambda_2 = 0 \tag{22}$$

$$-(\ln(w_x(\neg\text{bird}(x) \wedge \text{fly}(x)) + 1) + \lambda + \lambda_1 = 0 \tag{23}$$

$$-(\ln(w_x(\neg\text{bird}(x) \wedge \neg\text{fly}(x)) + 1) + \lambda = 0 \tag{24}$$

We rewrite these as follows,

$$w_x(\text{bird}(x) \wedge \text{fly}(x)) = e^{(\lambda-1)+\lambda_1+\lambda_2} = e^{(\lambda-1)}e^{\lambda_1}e^{\lambda_2} \tag{25}$$

$$w_x(\text{bird}(x) \wedge \neg\text{fly}(x)) = e^{(\lambda-1)+\lambda_2} = e^{(\lambda-1)}e^{\lambda_2} \tag{26}$$

$$w_x(\neg\text{bird}(x) \wedge \text{fly}(x)) = e^{(\lambda-1)+\lambda_1} = e^{(\lambda-1)}e^{\lambda_1} \tag{27}$$

$$w_x(\neg\text{bird}(x) \wedge \neg\text{fly}(x)) = e^{(\lambda-1)} = e^{\lambda-1} \tag{28}$$

Following Cheeseman, we abbreviate $e^{(\lambda-1)}$ by α_0, e^{λ_1} by α_1, and e^{λ_2} by α_2.

$$w_x(\text{bird}(x) \wedge \text{fly}(x)) = \alpha_0\alpha_1\alpha_2 \tag{29}$$

$$w_x(\text{bird}(x) \wedge \neg\text{fly}(x)) = \alpha_0\alpha_2 \tag{30}$$

$$w_x(\neg\text{bird}(x) \wedge \text{fly}(x)) = \alpha_0\alpha_1 \tag{31}$$

$$w_x(\neg\text{bird}(x) \wedge \neg\text{fly}(x)) = \alpha_0 \tag{32}$$

We could use the above three constraints to solve α_0, α_1, and α_2.

$$f_0(A(F)) = \alpha_0(\alpha_1\alpha_2 + \alpha_2 + \alpha_1 + 1) - 1 = 0 \tag{33}$$

$$f_1(A(F)) = \alpha_0\alpha_1\alpha_2 + \alpha_0\alpha_1 - p_2 = 0 \tag{34}$$

$$f_2(A(F)) = \alpha_0\alpha_1\alpha_2 J + \alpha_0\alpha_2 - p_3 = 0 \tag{35}$$

However, we use an alternative way of showing $p_1 = p_2$

$$\begin{aligned} p_1 &= w_x(\text{fly}(x)|\text{bird}(x)) \\ &= \frac{w_x(\text{bird}(x) \wedge \text{fly}(x))}{w_x(\text{bird}(x) \wedge \text{fly}(x)) + w_x(\text{bird}(x) \wedge \neg\text{fly}(x))} \\ &= \frac{\alpha_0\alpha_1\alpha_2}{\alpha_0\alpha_1\alpha_2 + \alpha_0\alpha_2} \\ &= \frac{\alpha_1}{\alpha_1 + 1} \end{aligned} \tag{36}$$

Furthermore, we have $p_2 = \alpha_0\alpha_1\alpha_2 + \alpha_0\alpha_1$. However, we have $\alpha_0(\alpha_1\alpha_2 + \alpha_2 + \alpha_1 + 1) = 1 \Leftrightarrow \alpha_0(\alpha_2 + 1)(\alpha_1 + 1) = 1 \Leftrightarrow \alpha_0(\alpha_2 + 1) = 1/(\alpha_1 + 1) \Leftrightarrow p_1 = p_2$.

In this example the maximum entropy completion was straightforward. Unfortunately, in general this does not seem to be the case. Quoting from [10], 'if we accept maximum entropy,..., the problem of actually computing weights to any reasonable approximation is NP-hard and thus probably infeasible'. While noting that using maximum entropy, 'yields patterns of reasoning that parallel common discourse', Pearl also warns that its biggest shortcoming is 'its computational complexity' ([11]).

In our approach the complexity problem occurs when setting the database up, but once the database is complete, then the reasoning is tractable. The price for this tractability is that the size of a database will grow exponentially with the number of predicates in the language. This is in contrast to existing non-monotonic logics where there is no analogue to completing the database, but the reasoning is intractable—for example for default logic [7]. We therefore see a trade-off between completing a database and reasoning with a database.

8 Discussion

We can summarize the advantages of our approach: (1) We only need probabilities on the domain to represent defeasible rules; and (2) we can use a 'natural' assumption to augment probability theory in a way to allow detachment of the consequents of such rules.

To expand the second point, probability logics do not capture all of the reasoning that is done in situations of uncertainty. In particular, we are interested in addressing issues of (1) choice of reference class, (2) typicality assumptions, and (3) taking decisions. All three involve default assumptions. Our primary aim is to formalise such default assumptions.

Our approach combines the notion of preferred models with that of Halpern type-1 semantics. First, if our database is incomplete, we select a subset of the models for that database to force a completion by maximising entropy. Second, for a completed database, we take a subset of the corresponding models to force the entailment of the default inferences.

Acknowledgements. This work is currently being funded by UK SERC grants GR/G 29861 and GR/G 29854 for the Rule-based Systems project. Special thanks are due to Dov Gabbay and Donald Gillies. We also are very grateful to anonymous referees for helpful suggestions on improving previous versions of this work.

References

1. Fahiem Bacchus. *Representing and Reasoning with Probabilistic Knowledge: A Logical Approach to Probabilities.* MIT Press, Cambridge, MA, 1990.
2. P. Cheeseman. A method of computing generalized bayesian probability values. In *IJCAI'83*, pages 198–202. Morgan Kaufmann, 1983.
3. James Cussens and Anthony Hunter. Using defeasible logic for a window on a probabilistic database: some preliminary notes. In *Symbolic and Qualitative Approaches for Uncertainty*, pages 146–152. Lecture Notes in Computer Science 548, Springer, 1991.
4. Dov Gabbay. Theoretical foundations for non-monotonic reasoning in expert systems. In K. R. Apt, editor, *Proceedings NATO Advanced Study Institute on Logics and Models of Concurrent Systems*, pages 439–457. Springer, 1985.
5. B. Grosof. Non-monotonicity in probabilistic reasoning. In K. R. Apt, editor, *Uncertainty in Artificial Intelligence 3*, pages 237–249. North-Holland, 1988.
6. Joseph Y. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46:311–350, 1990.
7. H. Kautz and B. Selman. Hard problems for simple default logics. *Artificial Intelligence*, 49:243–279, 1991.
8. B. Lewis. Approximating probability distributions to reduce storage requirements. *Information and Control*, 2:214–225, 1959.
9. N. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28:71–87, 1986.
10. J. Paris and A. Vencovská. A note on the inevitability of maximum entropy. *International Journal of Approximate Reasoning*, 4:183–223, 1990.
11. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, 1988.
12. Y. Shoham. *Reasoning about Change.* MIT Press, 1988.