Belief Revision through Forgetting Conditionals in Conditional Probabilistic Logic Programs

Anbu Yue [1] and **Weiru Liu** [1]

Abstract. In this paper, we present a revision strategy of revising a conditional probabilistic logic program (PLP) when new information is received (which is in the form of probabilistic formulae), through the technique of variable forgetting.We first extend the traditional forgetting method to forget a conditional event in PLPs. We then propose two revision operators to revise a PLP based on our forgetting method. By revision through forgetting, the irrelevant knowledge in the original PLP is retained according to the minimal change principle. We prove that our revision operators satisfy most of the postulates for probabilistic belief revision. A main advantage of our revision operators is that a new PLP is explicitly obtained after revision, since our revision operator performs forgetting a conditional event at the syntax level.

1 Introduction

Belief revision is concerned with how to revise an agent's current belief when new evidence is received, where this new evidence is assumed to have the highest priority. Any belief in the current belief set that is inconsistent with the evidence has to be weakened or omitted in order to get a revised consistent set of beliefs. In the literature of probabilistic belief revising, most research focuses on revising a single probabilistic distribution [5, 1, 8, 4, 3]. However, a single probabilistic distribution is not suitable for representing imprecise probabilistic beliefs, as the case for a conditional probabilistic logic program (PLP), where a set of probability distributions are usually associated with a PLP [13, 14]. Research on revising a set of probabilistic distributions are reported in [16, 7], but these methods (as well as methods on revising single probabilistic distributions) can only revise probability distributions by a certain kind of evidence, i.e., evidence that is consistent with the original distributions. Therefore, any evidence that is not fully consistent with current knowledge (beliefs) cannot be used.

The notion of *forgetting* (facts) (or referred to as *variable forgetting*) proposed in [12] has been applied (or adapted) in many logic based reasoning techniques. For example, forgetting is used for belief merging in [11], and the relationship between forgetting and belief change is studied in [15]. Traditionally, the main focus has been on forgetting a fact in classical logics. The issue of forgetting conditional knowledge has not been investigated, whilst conditional knowledge is very important, especially in research on (logic) reasoning with conditionals [13, 14].

In this paper, we extend the method of forgetting to *forget conditional events* in *conditional probabilistic logic programs* (PLPs). Given a PLP P, forgetting a conditional event $(\psi|\phi)$ in P means that fact ψ is forgotten only in the *domain* defined by ϕ. Assume that $(\psi'|\phi')[l', u'] \in P$, the challenge is how to retain part or all of knowledge $(\psi'|\phi')[l', u']$ when ϕ' and ϕ are inequivalent. To achieve this, we define a notion of *irrelevnace* for conditional events, so that forgetting a conditional event will retain any irrelevant knowledge. Since any classical theory T can be represent by a PLP [13], we prove that forgetting a fact ψ in a classical theory T is equivalent to forgetting a conditional event $(\psi|\top)$ in P that represents T.

Based on the technique of forgetting a conditional event, we propose two operators for revising PLPs by a probabilistic formula of the form $(\psi|\phi)[l, u]$. Our revision operators satisfy most of the postulates for imprecise probabilistic belief revision. These postulates were proposed in [18] and were proved to be an extension of Darwiche and Pearl postulates [2], Bayesian conditioning and Jeffrey's rule. Since any conditional event can be forgotten in a PLP, our revision operators do not require new evidence (information) to be consistent with the original PLPs. Another advantage of these revision operators is that, a new PLP is explicitly obtained as the result of revision, since forgetting a conditional event is defined at the syntax level. This is in contract to traditional probabilistic revision mentioned above where a revision result is a single or a set of probability distributions (which can be seen as the models of a probabilistic knowledge base, e.g., PLP).

This paper is organized as follows. In the next section, we briefly review probabilistic logic programming, postulates for probabilistic belief revision, and forgetting. In Section 3, we propose an approach to forgetting a conditional event in a PLP, and in Section 4, we propose two belief revision operators and give their properties. After comparing with related work in Section 5, we conclude this paper.

2 Preliminaries

2.1 Probabilistic logic programs (PLPs)

We briefly review conditional probabilistic logic programs here, see [13, 14] for details.

Let Φ be a finite set of *predicate symbols* and *constant symbols*, and $\mathcal{V}$ be a set of *object variables* and $\mathcal{B}$ be a set of *bound constants* which are in [0,1] describing the bound of probabilities. It is required that Φ contains at least one constant symbol. We use lowercase letters $a, b, \ldots$ for constants from Φ, uppercase letters X, Y for object variables, and l, u for bound constants. In Φ, there are two predicate symbols $\top$ and $\bot$ which represent *true* and *false* respectively.

An *object term* is a constant from Φ or an object variable from $\mathcal{V}$. An *atom* is of the form $p(t_1, \ldots, t_k)$, where p is a predicate symbol and t_i is an object term. An *event* or *formula* is constructed from

[1] School of Electronics, Electrical Engineering and Computer Science, Queen's University of Belfast, Belfast BT7 1NN, UK {a.yue, w.liu}@qub.ac.uk

a set of atoms by logic connectives $\wedge, \vee, \neg$ as usual, and a *conditional event* is of the form $\psi|\varphi$ with events ψ and φ. We use Greek letters ϕ, ψ, φ for events, α, β for conditional events. A *probabilistic formula* is of the form $(\psi|\varphi)[l, u]$ which means that the probability bounds for conditional event $\psi|\varphi$ are l and u. We call ψ its *consequent* and φ its *antecedent*. A *conditional probabilistic logic program (PLP) P* is a set of probabilistic formulae. We use $\mathcal{PL}$ to denote the set of all PLPs, and $\mathcal{F}$ to denote the set of all probabilistic formulas. An object term, event, conditional event, probabilistic formula, or PLP is called *ground* iff it does not contain any object variables from $\mathcal{V}$.

Herbrand universe (denoted as HU_Φ) is the set of all constants from Φ, and Herbrand base HB_Φ is a finite nonempty set of all events constructed from the predicate symbols in Φ and constants in HU_Φ. A *possible world* I is a subset of HB_Φ s.t. $\top \in I$ and $\bot \notin I$, and $\mathcal{I}_\Phi$ is the set of all possible worlds over Φ. An *assignment* σ maps each object variable to an element of HU_Φ. It is extended to object terms by $\sigma(c) = c$ for all constant symbols from Φ. An event φ is satisfied by I under σ, denoted by $I \models_\sigma \varphi$, is defined inductively as:

- $I \models_\sigma p(t_1, \ldots, t_n)$ iff $p(\sigma(t_1), \ldots, \sigma(t_n)) \in I$;
- $I \models_\sigma \phi_1 \wedge \phi_2$ iff $I \models_\sigma \phi_1$ and $I \models_\sigma \phi_2$;
- $I \models_\sigma \phi_1 \vee \phi_2$ iff $I \models_\sigma \phi_1$ or $I \models_\sigma \phi_2$;
- $I \models_\sigma \neg\phi$ iff $I \not\models_\sigma \phi$

An event φ is satisfied by a possible world I, or I is a *model* of φ, denoted by $I \models_{cl} \varphi$, iff $I \models_\sigma \varphi$ for all assignment σ. In this paper, we call the set of the models of φ the *domain* of φ. An event φ is a *logical consequence* of event ϕ, denoted as $\phi \models_{cl} \varphi$, iff all possible worlds that satisfy ϕ also satisfy of φ.

A *probabilistic interpretation* Pr is a probability distribution on $\mathcal{I}_\Phi$ (i.e., as $\mathcal{I}_\Phi$ is finite, Pr is a mapping from $\mathcal{I}_\Phi$ to the unit interval [0,1] such that $\sum_{I \in \mathcal{I}_\Phi} Pr(I) = 1$). The *probability* of an event φ in Pr under an assignment σ, is defined as $Pr_\sigma(\varphi) = \sum_{I \in \mathcal{I}_\Phi, I \models_\sigma \varphi} Pr(I)$. If φ is ground, we simply write as $Pr(\varphi)$.

A probabilistic formula $(\psi|\varphi)[l, u]$ is satisfied by a probabilistic interpretation Pr under an assignment σ, denoted by: $Pr \models_\sigma (\psi|\varphi)[l, u]$ iff $Pr_\sigma(\varphi) = 0$ or $Pr_\sigma(\psi|\varphi) \in [l, u]$.

A probabilistic formula μ is *satisfied* by a probabilistic interpretation Pr, or Pr is a *probabilistic model* of μ, denoted by $Pr \models \mu$, iff $Pr \models_\sigma \mu$ for all assignments σ. A probabilistic interpretation is a *probabilistic model* of a PLP P, denoted by $Pr \models P$, iff Pr is a probabilistic model of all $\mu \in P$. A PLP P is *satisfiable* or *consistent* iff a model of P exists. A probabilistic formula $(\psi|\varphi)[l, u]$ is a *consequence* of the PLP P, denoted by $P \models (\psi|\varphi)[l, u]$, iff all probabilistic models of P are also probabilistic models of $(\psi|\varphi)[l, u]$. A probabilistic formula $(\psi|\varphi)[l, u]$ is a *tight consequence* of P, denoted by $P \models_{tight} (\psi|\varphi)[l, u]$, iff $P \models (\psi|\varphi)[l, u]$, $P \not\models (\psi|\varphi)[l, u']$, $P \not\models (\psi|\varphi)[l', u]$ for all $l' > l$ and $u' < u$ ($l', u' \in [0, 1)$). Notice that, if $P \models (\phi|\top)[0, 0]$, then it is canonically defined as $P \models_{tight} (\psi|\phi)[1, 0]$, where $[1, 0]$ stands for an empty set.

2.2 Probabilistic belief revision

We briefly review the postulates for revising PLPs here, see [18] for details.

Given a PLP P, we define set $Bel^0(P)$ as $Bel^0(P) = \{(\psi|\phi)[l, u] \mid P \models (\psi|\phi)[l, u], P \not\models (\phi|\top)[0, 0]\}$ and call it the belief set of P. Condition $P \not\models (\phi|\top)[0, 0]$ is required because when $P \models (\phi|\top)[0, 0]$, $P \models (\psi|\phi)[l, u]$ for all ψ and all $[l, u] \subseteq [0, 1]$. Without this condition, some counterintuitive conclusions can be inferred, for instance, $(\psi|\phi)[0, 0.3]$ and $(\psi|\phi)[0.9, 1]$ can simultaneously be the beliefs of an agent if $P \models (\phi|\top)[0, 0]$.

Each *probabilistic epistemic state*, Ψ, has a unique belief set, denoted as $Bel^0(\Psi)$, which is a set of probabilistic formulae. $Bel^0(\Psi)$ is *closed*, i.e. $Bel^0(Bel^0(\Psi)) = Bel^0(\Psi)$. We call Ψ a probabilistic epistemic state of a PLP P, iff $Bel^0(\Psi) = Bel^0(P)$. In general, there exist many ways to define probabilistic epistemic state. e.g., we can define a probabilistic epistemic state as the set of probabilistic distributions that satisfies the PLP, see [18] for details.

Furthermore, we have the following inference relations:

$\Psi \models (\psi|\phi)[l, u]$ iff $(\psi|\phi)[l, u] \in Bel^0(\Psi)$, and

$\Psi \models_{tight} (\psi|\phi)[l, u]$ iff $\Psi \models (\psi|\phi)[l, u]$ and for all $[l', u'] \subset [l, u]$, $\Psi \not\models (\psi|\phi)[l, u]$.

We write $\Psi \wedge (\psi|\phi)[l, u]$ to represent $Bel^0(\Psi) \cup \{(\psi|\phi)[l, u]\}$. Also, $\Psi \models (\psi|\phi)[l, u]$ iff $P \models (\psi|\phi)[l, u]$ when $P \not\models_{tight} (\phi|\top)[0, 0]$.

Definition 1 *A conditional event $(\psi|\phi)$ is more specific than another conditional event $(\psi'|\phi')$, denoted as $(\psi|\phi) \trianglelefteq (\psi'|\phi')$, iff*

- *$\phi \models_{cl} \phi' \wedge \psi'$, or*
- *$\phi \models_{cl} \phi' \wedge \neg\psi'$.*

Conditional event $(\psi|\phi)$ affects only the relationship (probability distributions) between $\phi \wedge \psi$ and $\phi \wedge \neg\psi$. When $(\psi|\phi) \trianglelefteq (\psi'|\phi')$ holds, $(\psi|\phi)$ provides detailed information about ϕ, which is a sub-event of $\phi' \wedge \psi'$ or $\phi' \wedge \neg\psi'$. Therefore, $(\psi|\phi)$ is more specific than $(\psi'|\phi')$.

Definition 2 (perpendicular) *A conditional event $(\psi|\phi)$ is perpendicular with another conditional event $(\psi'|\phi')$, denoted as $(\psi|\phi) \bowtie (\psi'|\phi')$ iff $(\psi|\phi) \trianglelefteq (\psi'|\phi')$, or $(\psi'|\phi') \trianglelefteq (\psi|\phi)$, or $\models_{cl} \neg(\phi' \wedge \phi)$.*

The perpendicularity relation formalizes a kind of *irrelevance* between two conditional events. The above definition is an extension of the definition of perpendicular in [9], in which the first condition is not required. If $(\psi|\phi) \trianglelefteq (\psi'|\phi')$, then $(\psi|\phi)$ is more specific than $(\psi'|\phi')$ and thus $(\psi|\phi)$ will not affect $(\psi'|\phi')$. We know that $(\psi|\phi)$ can not affect the probability distributions *within* the domain $(\psi \wedge \phi)$ or the domain $(\neg\psi \wedge \phi)$, so if $(\psi'|\phi') \trianglelefteq (\psi|\phi)$, then ϕ' is a sub-event of $(\psi \wedge \phi)$ or $(\neg\psi \wedge \phi)$, and therefore $(\psi|\phi)$ can not affect $(\psi'|\phi')$. If $\models_{cl} \neg(\phi' \wedge \phi)$, then ϕ and ϕ' have disjoint domains, so $(\psi|\phi)$ and $(\psi'|\phi')$ are irrelevant.

Definition 3 ([18]) *Let P be a PLP with epistemic state Ψ and $\mu = (\psi|\phi)[l, u]$ be a probabilistic formula. The result of revising P by μ is another probabilistic epistemic state, denoted as $\Psi \star \mu$ where $\star$ is a* revision operator. *Operator $\star$ is required to satisfy the following postulates:*

R*1 $\Psi \star \mu \models \mu$
R*2 $\Psi \wedge \mu \models \Psi \star \mu$
R*3 *if $\Psi \wedge \mu$ is satisfiable, then $\Psi \star \mu \models \Psi \wedge \mu$*
R*4 *$\Psi \star \mu$ is unsatisfiable only if μ is unsatisfiable*
R*5 *$\Psi \star \mu \equiv \Psi \star \mu'$ if $\mu \equiv \mu'$*
R*6 *Let $\mu = (\psi|\phi)[l, u]$ and $\Psi \star \mu \models_{tight} (\psi|\phi)[l', u']$. Let $\mu' = (\psi|\phi)[l_1, u_1]$ and $\Psi \star \mu' \models_{tight} (\psi|\phi)[l_1', u_1']$. For any $\epsilon > 0$, if $|u_1 - u| + |l_1 - l| < \epsilon$, and both of $(\psi|\phi)[l, u]$ and $(\psi|\phi)[l', u']$ are satisfiable, then $|u_1' - u'| + |l_1' - l'| < \epsilon$.*
R*7 *if $\Psi \models (\phi|\top)[l, u]$, then $(\Psi \star \mu) \models (\phi|\top)[l, u]$*
R*8 *Let $\mu = (\psi|\phi)[l, u]$ and $\mu' = (\psi'|\phi')[l', u']$. Suppose that $(\psi|\phi) \bowtie (\psi'|\phi')$. If $(\Psi \star \mu) \wedge \mu'$ is satisfiable then $\Psi \wedge \mu'$ is satisfiable, and $(\Psi \star \mu) \wedge \mu' = (\Psi \wedge \mu') \star \mu$.*

R*1 - R*5 is an analog to postulates R1 - R4 in [2]. We do not have corresponding postulates for R5 and R6 in [2] since revision with the conjunction of conditional events are more complicated and is beyond the scope of this paper. R*6 is a *sensitivity* requirement, which says that a slightly modification on the bounds of $\mu = (\psi|\phi)[l, u]$ (i.e., $\mu' = (\psi|\phi)[l_1, u_1]$) shall not affect the result of revision significantly. R*7 says that revising by $\mu = (\psi|\phi)[l, u]$ should not affect the statement about ϕ (but the impreciseness of ϕ may be decreased). Recall that perpendicular condition characterizes a kind of irrelevance, R*8 says that any irrelevance knowledge with new evidence should not be affected by the revision using this evidence.

It is proved that these postulates is an extension of modified AGM postulates and Darwiche and Pearl postulates for iterative revision [2]. It is also proved that these postulates lead to Jeffrey's rule and Bayesian conditioning when the original PLP (probabilistic epistemic) defines a single probability distribution.

2.3 Forgetting a fact

Given a set of ground formulas T and an atom p, forgetting p in T means obtaining another set of formulas which is weaker than T, but retain the same conclusions that *irrelevant* to p. Let $p(\vec{t})$ be a ground atom, and I_1, I_2 be two possible worlds. Define $I_1 \approx_{p(\vec{t})} I_2$ iff I_1 and I_2 agree on everything except possibly on the truth value of $p(\vec{t})$:

1. I_1 and I_2 have the same domain, i.e. I_1 and I_2 are defined on the same Herbrand base.
2. for every predicate symbol q that differs from p, and for every ground term $\vec{t}'$, $q(\vec{t}') \in I_1$ iff $q(\vec{t}') \in I_2$.

Definition 4 ([12]) *Let T be a set of formulae and $p(\vec{t})$ be a ground atom. The result of* forgetting *$p(\vec{t})$, denoted as $T' = forget_{cl}(T, p(\vec{t}))$, is a set of formulae such that, for any possible world I', I' is a model of T' iff there is a model I of T such that $I \approx_{p(\vec{t})} I'$.*

Proposition 1 ([12]) *For any theory T and ground atom $p(\vec{t})$, $T \models forget_{cl}(T, p(\vec{t}))$.*

Let φ be a ground formula and $p(\vec{t})$ be a ground atom. We use $\varphi^+_{p(\vec{t})}$ (resp. $\varphi^-_{p(\vec{t})}$) to denote the result of replacing every occurrence of $p(\vec{t})$ in φ by $\top$ (resp. $\bot$).

Proposition 2 ([12]) *Let φ be a ground formula and $p(\vec{t})$ be a ground atom. Suppose that theory $T = \{\varphi\}$, then $forget_{cl}(T, p(\vec{t})) \equiv \{\varphi^+_{p(\vec{t})} \vee \varphi^-_{p(\vec{t})}\}$.*

Let $p_1(\vec{t_1}), \ldots, p_n(\vec{t_n})$ be a sequence of ground atoms. The result of *forgetting* $p_1(\vec{t_1}), \ldots, p_n(\vec{t_n})$ in T, denoted as $forget_{cl}(T, p_1(\vec{t_1}), \ldots, p_n(\vec{t_n}))$, is inductively defined as $forget_{cl}(forget_{cl}(T, p_1(\vec{t_1}), \ldots, p_{n-1}(\vec{t}_{n-1})), p_n(\vec{t_n}))$.

Proposition 3 ([12]) *For any theory T and any ground atoms $p_1(\vec{t_1}), p_2(\vec{t_2})$, $forget_{cl}(forget_{cl}(T, p_1(\vec{t_1})), p_2(\vec{t_2}))$ and $forget_{cl}(forget_{cl}(T, p_2(\vec{t_2})), p_1(\vec{t_1}))$ are logically equivalent.*

The above proposition indicates that the order of the sequence $p_1(\vec{t_1}), \ldots, p_n(\vec{t_n})$ is not important in $forget_{cl}(T, p_1(\vec{t_1}, \ldots, p_n(\vec{t_n})))$. In this paper, we write $forget_{cl}(T, \mathcal{A})$ to represent $forget_{cl}(T, p_1(\vec{t_1}), \ldots, p_n(\vec{t_n}))$, where $\mathcal{A} = \{p_1(\vec{t_1}), \ldots, p_n(\vec{t_n})\}$. We also write $forget_{cl}(T, \phi)$ to represent $forget_{cl}(T, \mathcal{A}_\phi)$, where $\mathcal{A}_\phi$ is the set of atoms that appear in ϕ.

3 Forgetting a Conditional Event

Sometimes, forgetting a fact under certain conditions is useful, for example, forgetting fact φ when ϕ is given. To achieve this, we provide an approach to forgetting a conditional event $(\psi|\phi)$, which means forgetting ψ only in the domain of ϕ, and keeping the original knowledge that is out of the domain of ϕ unchanged.

Definition 5 *Let $[l, u]$ and $[l', u']$ be two intervals. The closest sub-interval of $[l', u']$ to $[l, u]$, denoted as $clb([l', u'], [l, u])$, is defined by $clb([l', u'], [l, u]) = [l_b, u_b]$, where*

- *if $u' < l$ then $l_b = u_b = u'$,*
- *if $l' > u$ then $l_b = u_b = l'$,*
- *otherwise, $l_b = \max\{l, l'\}, u_b = \min\{u, u'\}$.*

Definition 6 *Let P be a PLP and $\mu \in P$ where $\mu = (\psi_1|\phi_1)[l, u]$. Assume that $\nu = (\psi_2|\phi_2)$ is a conditional event. We define $forget_P(\mu, \nu)$ as:*

$$forget_P(\mu, \nu) = \left\{ \begin{array}{l} (\phi_2|\phi_1)[l_a, u_a], (\phi_1|\phi_2)[l_b, u_b], \\ (\psi_1|\phi_1 \wedge \neg\phi_2)[l_1, u_1], \\ (forget_{cl}(\psi_1, \psi_2)|\phi_1 \wedge \phi_2)[l_2, u_2] \end{array} \right\}$$

where

$$\begin{array}{l} P \models_{tight} (\phi_2|\phi_1)[l_a, u_a], P \models_{tight} (\phi_1|\phi_2)[l_b, u_b], \\ P \models_{tight} (\psi_1|\phi_1 \wedge \neg\phi_2)[l', u'], P \models_{tight} (\psi_1|\phi_1)[l", u"], \\ clb([l', u'], [l", u"]) = [l_1, u_1], \\ P \models_{tight} (forget_{cl}(\psi_1, \psi_2)|\phi_1 \wedge \phi_2)[l_2, u_2]. \end{array}$$

We define $forget(P, \nu) = \bigcup_{\mu \in P} forget_P(\mu, \nu)$.

When forgetting a conditional event $(\psi_2|\phi_2)$, the domain of the original beliefs should be divided into two parts: within the domain of ϕ_2 and out of the domain of ϕ_2. That is, if $(\psi_1|\phi_1)[l, u] \in P$, then the knowledge about $(\psi_1|\phi_1)$ in P is implicitly contained by $(\psi_1|\phi_2 \wedge \phi_1)$ and $(\psi_2|\phi_1 \wedge \neg\phi_2)$. Intuitively, the former may be affected and the latter should be retained. Also, the knowledge about $(\psi_1|\phi_1)$ should be changed as minimal as possible. To achieve this, the knowledge about $(\psi_1|\phi_1)$ must be retained by the knowledge about $(\psi_1|\phi_1 \wedge \neg\phi_2)$ in the result PLP. In addition, the relationships (subsumption, overlap, disjoint, etc.) between the domains of ϕ_1 and ϕ_2 should not be affected.

Proposition 4 *Let P be a PLP, and $\nu = (\psi|\phi)$ be a conditional event. If $P \not\models (\phi|\top)[0, 0]$ then $forget(P, \nu) \models_{tight} \nu[0, 1]$. If $P \models (\phi|\top)[0, 0]$ then $forget(P, \nu) \equiv P$, and we have that $\nu \notin forget(P, \nu)$.*

In the above proposition, $P \models (\phi|\top)[0, 0]$ indicates that any conditional event with ϕ as the antecedent has no effects on the semantics of P, however, at the syntax level $\nu \notin forget(P, \nu)$.

Proposition 5 *Let $P = \{(\psi_1|\phi_1)[l_1, u_1], \ldots, (\psi_n|\phi_n)[l_n, u_n]\}$ be a PLP, and $\nu = (\psi|\phi)$ be a conditional event. Suppose that $(\psi|\phi) \trianglelefteq (\psi_i|\phi_i)$ for all $i \in \{1, \ldots, n\}$, then $forget(P, \nu) \equiv P$.*

However, if $P = \{(\psi_1|\phi_1)[l_1, u_1], \ldots, (\psi_n|\phi_n)[l_n, u_n]\}$ and $(\psi_i|\phi_i) \trianglelefteq (\psi|\phi)$ holds for $i = 1, ...n$, then $forget(P, (\psi|\phi)) \equiv P$ does not hold in general. This is because that forgetting a conditional event $(\psi|\phi)$ will forget not only the relationship between $(\phi \wedge \psi)$ and $(\phi \wedge \neg\psi)$, but also all statements about ψ in the domain of ϕ.

Proposition 6 *Let $P = \{(\phi_1 \wedge \cdots \wedge \phi_n|\top)[1,1]\}$ and $\nu = (\varphi|\top)$. Then for any event ψ, $forget(P,\nu) \models (\psi|\top)[1,1]$ iff $forget_{cl}(\{\phi_1 \wedge \cdots \wedge \phi_n\},\varphi) \models_{cl} \psi$.*

Let two theories be $T_1 = \{\phi_1, \ldots, \phi_n\}$ and $T_2 = \{\phi_1 \wedge \cdots \wedge \phi_n\}$, then $T_1 \equiv T_2$ and T_2 is logically equivalent to PLP $P = \{(\phi_1 \wedge \cdots \wedge \phi_n|\top)[1,1]\}$, $forget(P,\nu)$ is equivalent to $forget_{cl}(T,\varphi)$, where $\nu = (\varphi|\top)$. As a consequence, forgetting facts is a special case of forgetting conditional events.

Definition 7 *Let P be a PLP and its set of probability distributions be $\mathbf{Pr}$, and $\nu = (\psi|\phi)$ be a conditional event. We let $\mathbf{Pr}_P^\nu$ be the set of probabilistic distributions s.t. $Pr' \in \mathbf{Pr}_P^\nu$ iff there exists a $Pr \in \mathbf{Pr}$ such that*

(1) $Pr'(I) = Pr(I)$, *if* $I \not\models \phi$

(2) $\sum_{J \models \phi, J \approx_\psi I} Pr'(J) = \sum_{J \models \phi, J \approx_\psi I} Pr(J)$, *if* $I \models \phi$

(3) $Pr'(\phi \wedge \phi') = Pr(\phi \wedge \phi')$, *if there exists* $(\psi'|\phi')[l,u] \in P$

In the above definition, condition (1) means that when ϕ is not satisfied, then nothing should be forgotten; condition (2) says that even when ϕ is satisfied, only the beliefs that are relevant to ψ are forgotten; condition (3) says that within the domain of ϕ, the probabilities of the antecedents of probabilistic formulae in P should not be affected.

Obviously, $\mathbf{Pr} \subseteq \mathbf{Pr}_P^\nu$ and therefore, $\mathbf{Pr}_P^\nu$ is not empty iff P is satisfiable.

Proposition 7 *Let P be a PLP and $\nu = (\psi|\phi)$ be a conditional event. Then $\mathbf{Pr}_P^\nu$ is the set of probabilistic models of $forget(P,\nu)$.*

Forgetting a conditional event will not loss knowledge that is irrelevant to the forgotten conditional event.

Proposition 8 *Let P be a PLP and $\nu = (\psi|\phi)$ be a conditional event. If $P \models (\psi'|\phi')[l,u]$ then $forget(P,\nu) \models (\psi'|\phi')[l,u]$, where $(\psi|\phi) \trianglelefteq (\psi'|\phi')$ or $\models_{cl} \neg(\phi \wedge \phi')$.*

Example 1 *Let P be given as:*

$$P = \left\{ \begin{array}{l} (fly(t)|bird(t))[0.98,1] \\ (bird(t)|penguin(t))[1,1] \\ (penguin(t)|bird(t))[0.1,1] \end{array} \right\}$$

From P, it can be inferred that $P \models (fly(t)|penguin(t))[0.8,1]$. When it is informed that this conclusion may be wrong, we want to revise P by forgetting $\nu = (fly(t)|penguin(t))$. After forgetting ν from P we can get the PLP $forget(P,\nu)$. It is worth noting that, for any PLP P', for any events ϕ and ψ and any $l,u \in [0,1]$, the statements $P' \models (\top|\phi)[1,1]$, $P' \models (\phi|\bot)[l,u]$ and $P' \models (\psi|\phi \wedge \psi)[1,1]$ always hold. By omitting such kind of probabilistic formulae, $forget(P,\nu)$ can be simplified as:

$$forget(P,\nu) = \left\{ \begin{array}{l} (penguin(t)|bird(t))[0.1,1] \\ (bird(t)|penguin(t))[1,1] \\ (fly(t)|bird(t) \wedge \neg penguin(t))[0.98,1] \end{array} \right\}$$

In the original P, $P \models_{tight} (fly(t)|bird(t) \wedge \neg penguin(t))[0,1]$. The lower bound is from the assumption that it is possibly that all birds are penguins and all penguins cannot fly. In another word, this conclusion depends on the knowledge about $(fly(t)|penguin(t))$ which should be forgotten, and thus, this bound is not suitable. On the contrary, it is stated in $forget(P,\nu)$ that $(fly(t)|bird(t) \wedge \neg penguin(t))[0.98,1]$, which retains the knowledge that a bird (which is not a penguin) very likely can fly. Let $P' = forget(P,\nu)$, we have $P' \models (fly(t)|penguin(t))[0,1]$, which means that indeed in P', the knowledge about whether penguins can fly is totally forgotten.

4 Belief Revision by Forgetting

In this section, we define two specific revision operators that revising a PLP P with a probabilistic formula.

Definition 8 *Let P be a PLP in $\mathcal{PL}$, and $\mu = (\psi|\phi)[l,u]$ be a probabilistic formula in $\mathcal{F}$. Let $\nu = (\psi|\phi)$. We define operator $\diamond_0 : \mathcal{PL} \times \mathcal{F} \rightarrow \mathcal{F}$ such that $P \diamond_0 \mu = forget(P,\nu) \cup \{\mu\}$.*

Example 2 *Let P be given as in Example 1, and $\mu = (fly(t)|penguin(t))[0,0]$ be a probabilistic formula.*

$$P \diamond_0 \mu = \left\{ \begin{array}{l} (fly(t)|bird(t) \wedge \neg penguin(t))[0.98,1] \\ (bird(t)|penguin(t))[1,1] \\ (penguin(t)|bird(t))[0.1,1] \\ (fly(t)|penguin(t))[0,0] \end{array} \right\}$$

Now we can infer that $P \diamond_0 \mu \models_{tight} (fly(t)|bird(t))[0,0.9]$ which is intuitively correct. The upper bound of the probability if whether a bird can fly is changed to be 0.9 following the fact that some birds (penguins) cannot fly.

In the above example, we have $P \diamond_0 \mu \models_{tight} (fly(t)|bird(t))[0,0.9]$. This lower bound (0) means that it is possible that all birds cannot fly. The lower bound comes from the possibility that all birds are penguins since $P \models (penguin(t)|bird(t))[0.1,1]$. Using operator $\diamond_0$ to revise P with $(fly(t)|penguin(t))[0,0]$ does not eliminate such possibility.

On the another hand, since the new information that penguins cannot fly contradicts with the original general knowledge that most birds can fly, it implicitly suggests that penguins are very different from typical birds. Formally, the probability of $(penguin(t)|bird(t))$ should be low. In fact, if we had $(penguin(t)|bird(t))[0.1,0.1]$ in $P \diamond_0 \mu$ in the above example, we should have got $P' = (P \diamond_0 \mu) \cup \{(penguin(t)|bird(t))[0.1,0.1]\}$ and $P' \models_{tight} (fly(t)|bird(t))[0.882,0.9]$ which gives a much tighter and more intuitive bounds for $(fly(t)|bird(t))$.

This discussion suggests that sometimes the contradiction between new information $(\psi|\phi)[l,u]$ and an original PLP P implies that the antecedent ϕ is a special case of ϕ' for any ϕ' that $(\psi'|\phi')[l',u'] \in P$ and ϕ' is relevant to $(\psi|\phi)$. Here ϕ' is relevant to $(\psi|\phi)$ means that a tighter probability bound for $(\psi|\phi)$ can be inferred from P only when more knowledge about the relationship between ϕ' and ϕ (i.e. a tighter bound for $(\phi'|\phi)$ or $(\phi|\phi')$) is provided. The above discussion leads us to define another revision operator $\diamond$. Revising with this operator, the impreciseness of the antecedent of new information may be decreased.

Definition 9 *Let P be a PLP in $\mathcal{PL}$, and $\mu = (\psi|\phi)[l,u]$ be a probabilistic formula in $\mathcal{F}$. Let $\nu = (\psi|\phi)$. We define operator $\diamond : \mathcal{PL} \times \mathcal{F} \rightarrow \mathcal{PL}$ which satisfies*

(1) $\mu \in P \diamond \mu$

(2) $forget(P,\nu) \subseteq P \diamond \mu$

(3) $\forall (\psi'|\phi')[l,u] \in P$,
$(\phi'|\phi)[l_a,u_a] \in P \diamond \mu$ *and* $(\phi|\phi')[l_b,u_b] \in P \diamond \mu$

where

$P \models_{tight} (\psi|\phi)[l_0,u_0]$

$clb([l_0,u_0],[l,u]) = [l',u']$

$P \cup \{(\psi|\phi)[l',u']\} \models_{tight} (\phi'|\phi)[l_a,u_a]$

$P \cup \{(\psi|\phi)[l',u']\} \models_{tight} (\phi|\phi')[l_b,u_b]$

and $P \diamond \mu$ is the smallest set (with respect to set inclusion) that satisfying the above conditions.

Obviously, $P \diamond \mu \models P \diamond_0 \mu$.

Example 3 *Let P be as given in Example 1, and $\mu = (fly(t)|penguin(t))[0,0]$ be a probabilistic formula.*

$$P \diamond \mu = \left\{ \begin{array}{l} (fly(t)|bird(t) \wedge \neg penguin(t))[0.98, 1] \\ (bird(t)|penguin(t))[1, 1] \\ (penguin(t)|bird(t))[0.1, 0.1] \\ (fly(t)|penguin(t))[0, 0] \end{array} \right\}$$

Now, we have that most birds can fly since $P \diamond \mu \models_{tight} (fly(t)|bird(t))[0.882, 0.9]$ and this knowledge is still imprecise.

Proposition 9 *Both operators $\diamond_0$ and $\diamond$ satisfy the postulates R*1, R*2, R*4, R*6, and R*7.*

Both operators do not satisfy R*3 in general. This comes from the fact that our operators retain the impreciseness of the original knowledge whilst $\Psi \wedge \mu$ decreases the impreciseness of the original knowledge.

These two operators also do not satisfy R*8 in general. From R*8, we can get three weaker postulates:

R*8′.1 for all ψ' and ϕ', if $(\psi|\phi) \trianglelefteq (\psi'|\phi')$ and $P \star (\psi|\phi)[l,u] \models (\psi'|\phi')[l',u']$ then $P \models (\psi'|\phi')[l',u']$.
R*8′.2 for all ψ' and ϕ', if $(\psi'|\phi') \trianglelefteq (\psi|\phi)$ and $P \star (\psi|\phi)[l,u] \models (\psi'|\phi')[l',u']$ then $P \models (\psi'|\phi')[l',u']$.
R*8′.3 for all ψ' and ϕ', if $\models_{cl} \neg(\phi \wedge \phi')$ and $P \star (\psi|\phi)[l,u] \models (\psi'|\phi')[l',u']$ then $P \models (\psi'|\phi')[l',u']$.

Proposition 10 *The operator $\diamond_0$ and $\diamond$ satisfy R*8′.1 and R*8′.3.*

R*8′.2 is not satisfied by $\diamond_0$ and $\diamond$ because forgetting conditional event $(\psi|\phi)$ may affect the knowledge about $(\psi'|\phi')$ if $(\psi'|\phi') \trianglelefteq (\psi|\phi)$.

5 Related Work and Conclusion

Related work: Traditionally, forgetting is to delete some concepts (atoms or facts) from a given theory in a classical logic-based language. In this paper, we extended the concept of forgetting to forget conditional events other than facts in the framework of conditional probabilistic logic programming. Since facts can be represented as a special kind of conditional events, i.e., conditional events that have tautologies as its antecedent, it is not surprising that our forgetting method subsumes the original approach to forgetting facts.

In [15], forgetting facts is deployed in belief change in propositional logic. When reducing forgetting conditional events operation to forgetting facts in our operator $\diamond_0$ (since when the bounds for every probabilistic formula is either [0,0] or [1,1], a PLP actually contains a set of propositional formulae), we can obtain the update operator defined in [15]. However, there is no counterpart of our $\diamond$ in [15].

In the literature of probabilistic belief revision, most revision operators are model-based, that is a revision operator revises a single or a set of probability distributions, and the result is also a single or a set of probability distributions. This kind of revision makes the probabilistic knowledge implicit, especially when this knowledge is in the form of PLP. On the contrary, our operators are defined at the syntax level, and a revised PLP is obtained as the result.

Many probabilistic belief revision operators require that new knowledge is consistent with the original knowledge [1, 8, 3, 4, 10, 17]. In contrast, since any conditional event can be forgotten from a PLP, we do not require that new knowledge is consistent with a given PLP. Furthermore, our revision results can still be imprecise (See Example 3) while some other revision operators [1, 8, 3, 4, 5, 6, 10], produce single probability distributions as the result of revision.

Conclusions: In this paper, we extended the concept of forgetting to forgetting conditional events in PLPs and proposed two revision operators based on our forgetting (of conditional events) approach. Our revision operators forget inconsistent knowledge and retain irrelevant knowledge with respect to new information.

Among the two operators we have defined, the second operator ($\diamond$) is particularly designed for situations where the antecedent of a conditional event (new information) in the original PLP is imprecise. The first revision operator does not change anything (bounds of probabilities) about the antecedent after revision whilst the second operator decreases the imprecision of the antecedent (in terms of probability bounds). The rational of operator $\diamond$ comes from the assumption that if new information contradicts with the original PLP, then it suggests that the antecedent may be a special case of a general concept defined in this PLP (such as penguin is a special type of bird, but not a common type of bird).

Our operators satisfy most of the postulates for probabilistic belief revision and operate at the syntax level of a PLP, so that a new PLP is explicitly returned as the result of revision.

REFERENCES

[1] Hei Chan and Adnan Darwiche, ‘On the revision of probabilistic beliefs using uncertain evidence’, *Artif. Intell.*, **163**(1), 67–90, (2005).
[2] Adnan Darwiche and Judea Pearl, ‘On the logic of iterated belief revision’, *Artif. Intell.*, **89**(1-2), 1–29, (1997).
[3] Didier Dubois and Henri Prade, ‘Focusing vs. belief revision: A fundamental distinction when dealing with generic knowledge’, in *Proc. of ECSQARU-FAPR’97*, pp. 96–107, (1997).
[4] B. Van Fraasen, ‘Probabilities of conditionals’, in *Proc. of Foundations of Probability Theory, Statistical Inference, and Statistical Theories of Science*, pp. 261–300, (1976).
[5] I. R. Goodman and Hung T. Nguyen, ‘Probability updating using second order probabilities and conditional event algebra’, *Inf. Sci.*, **121**(3-4), 295–347, (1999).
[6] Adam J. Grove and Joseph Y. Halpern, ‘Probability update: Conditioning vs. cross-entropy’, in *Proc. of UAI’97*, pp. 208–214, (1997).
[7] Adam J. Grove and Joseph Y. Halpern, ‘Updating sets of probabilities’, in *Proc. of UAI’98*, pp. 173–182, (1998).
[8] Peter Grünwald and Joseph Y. Halpern, ‘Updating probabilities’, *J. Artif. Intell. Res. (JAIR)*, **19**, 243–278, (2003).
[9] Gabriele Kern-Isberner, ‘Postulates for conditional belief revision’, in *Proc. of IJCAI’99*, pp. 186–191, (1999).
[10] Gabriele Kern-Isberner and Wilhelm Rödder, ‘Belief revision and information fusion on optimum entropy’, *Int. J. Intell. Syst.*, **19**(9), 837–857, (2004).
[11] Jérôme Lang, Paolo Liberatore, and Pierre Marquis, ‘Propositional independence: Formula-variable independence and forgetting’, *J. Artif. Intell. Res. (JAIR)*, **18**, 391–443, (2003).
[12] Fangzhen Lin and Raymond Reiter, ‘Forget it!’, in *Working Notes, AAAI Fall Symposium on Relevance*, eds., Russell Greiner and Devika Subramanian, pp. 154–159, Menlo Park, California, (1994). American Association for Artificial Intelligence.
[13] Thomas Lukasiewicz, ‘Probabilistic logic programming.’, in *Proc. of ECAI’98*, pp. 388–392, (1998).
[14] Thomas Lukasiewicz, ‘Probabilistic logic programming with conditional constraints.’, *ACM Trans. Comput. Log.*, **2**(3), 289–339, (2001).
[15] Abhaya C. Nayak, Yin Chen, and Fangzhen Lin, ‘Forgetting and knowledge update’, in *Proc. of Australian Conference on Artificial Intelligence*, pp. 131–140, (2006).
[16] Damjan Skulj, ‘Jeffrey’s conditioning rule in neighbourhood models’, *Int. J. Approx. Reasoning*, **42**(3), 192–211, (2006).
[17] Frans Voorbraak, ‘Probabilistic belief change: Expansion, conditioning and constraining’, in *Proc. of UAI’99*, pp. 655–662, (1999).
[18] Anbu Yue and Weiru Liu, ‘Revising imprecise probabilistic beliefs in the framework of probabilistic logic programming’, in *Proc. of AAAI’08*, (2008).