

Belief Logic Programming: Uncertainty Reasoning with Correlation of Evidence^{*}

Hui Wan

Michael Kifer

State University of New York at Stony Brook
Stony Brook, NY 11794, USA

Abstract. Belief Logic Programming (BLP) is a novel form of quantitative logic programming in the presence of uncertain and inconsistent information, which was designed to be able to combine and correlate evidence obtained from non-independent information sources. BLP has non-monotonic semantics based on the concepts of belief combination functions and is inspired by Dempster-Shafer theory of evidence. Most importantly, unlike the previous efforts to integrate uncertainty and logic programming, BLP can correlate structural information contained in rules and provides more accurate certainty estimates. The results are illustrated via simple, yet realistic examples of rule-based Web service integration.

1 Introduction

Quantitative reasoning has been widely used for dealing with uncertainty and inconsistency in knowledge representation, and, more recently, on the Semantic Web. A less explored issue in quantitative reasoning is combining correlated pieces of information. Most works disregard correlation or assume that all the information sources are independent. Others make an effort to take some forms of correlation into account, but only in an ad hoc manner.

Among the models of uncertainty, probabilistic logic programming is particularly popular: [4, 13, 14, 17, 19, 20]—just to name a few. However, when these approaches are used to combine evidence from different sources, the usage of probabilistic models becomes questionable, as discussed in Section 6. Another well-established way of dealing with uncertainty is Fuzzy Logic [29]. It has been successful in many application domains, but remains controversial due to some of its properties [6, 8]. For example, if S' is a complement of the fuzzy set S , then $S \cap S' \neq \emptyset$ and even $S \subset S'$ are possible. This property is problematic for some applications.

Dempster-Shafer theory of evidence [5, 22] has also been central to many approaches to quantitative reasoning [1–3, 15, 21, 23, 27]. This theory is based on belief functions [22]—a generalization of probability distributions, which represents degrees of belief in various statements. If these beliefs come from different

^{*} This work is part of the SILK (Semantic Inference on Large Knowledge) project sponsored by Vulcan, Inc.

sources, the belief functions must be combined in order to obtain more accurate information. The difficult problem here is that these sources might not be independent.

Yet another line of work is based on deductive database methodology without dedicating to any particular theory of modeling uncertainty [10–12, 15, 24].

To the best of our knowledge, all existing works avoid correlating belief derivation paths, which often leads to counter-intuitive behavior when such correlation is essential for correctness. Most approaches simply restrict logic dependencies to avoid combining sources that are not independent [10, 23]. Those that do not, might yield incorrect or inaccurate results when combining sources that are correlated due to overlapping belief derivation paths. For example, consider two rules $A :- B \wedge C$ and $A :- B \wedge D$, each asserting its conclusion with certainty 0.5. The approach in [15] would directly combine the certainty factors for A derived from the two rules as if they are independent, assigning A a combined certainty, which is likely going to be too high. Clearly, the independence assumption does not hold here, as both rules rely on the same fact B . A few notable exceptions include Baldwin’s [1, 2], Lakshmanan’s [12] and Kersting’s [9] approaches. Kersting et al. provide a very general framework, which could, in principle, be used to handle correlation. However, combination of two inconsistent conclusions is hard to explain in probability theory. Both Baldwin’s and Lakshmanan’s methods assume that every pair of rules with the same head have the same correlation. Consequently their methods are inadequate for scenarios such as the one described in Sections 2 and 7.

This paper introduces a novel form of quantitative reasoning, called *Belief Logic Programming* (BLP). BLP was designed specifically to account for correlation of evidence obtained from non-independent and, possibly, contradictory information sources. BLP has non-monotonic semantics based on belief combination functions and inspired by Dempster-Shafer theory of evidence. The BLP theory is orthogonal to the choice of a particular method of combining evidence and, in fact, several different methods can be used simultaneously for different pieces of uncertain information.¹ Most importantly, unlike the previous efforts to integrate uncertainty and logic programming, BLP can correlate structural information contained in rules and provides more accurate certainty estimates. The framework and the results are illustrated using simple, yet realistic examples of rule-based integration of Web services that deal with uncertain information.

This paper is organized as follows. Section 2 presents a motivating example, which is revisited in Section 7 from a technical standpoint. Section 3 provides background on Dempster-Shafer theory of evidence. Then, we define the syntax of BLP in Section 4 and the semantics in Section 5. Section 6 discusses several aspects of BLP and relates it to some other theories of non-monotonic reasoning. Section 8 concludes the paper.

¹ The reader should not confuse Dempster-Shafer theory of evidence with Dempster’s combination rule. BLP does not depend on that combination rule, but can use it for modeling beliefs.

2 Motivating Example

A group of Stony Brook students is planning a trip to see a Broadway musical. Normally, it takes 1.5 hours by car to get to Manhattan, but the students know that Long Island Expressway is not called by the locals “the longest parking lot in America” for nothing. The students consult a traffic service, which integrates information from several independent information sources to provide traffic advisory along various travel routes. Let us assume that these sources are:

- weather forecast (rain, snow, fog)
- social activity (parades, motorcades, marathons)
- police activity (accidents, emergencies)
- roadwork

The service uses the following rules (which are simplified for this example) to generate advisories:

1. If the weather is bad, and there is roadwork along the route, the likelihood of a delay is 0.9.
2. If there is roadwork and social activities along the route, the likelihood of a delay is 0.8.
3. If there is roadwork and police activity along the route, the likelihood of a delay is 0.99.

These rules are expressed in BLP as shown below, where $?r$ is the variable that represents the travel route.

$$\begin{aligned} [0.9, 1] \quad \text{delay}(?r) & :- \text{roadwork}(?r) \wedge \text{bad_weather}(?r) \\ [0.8, 1] \quad \text{delay}(?r) & :- \text{roadwork}(?r) \wedge \text{social_act}(?r) \\ [0.99, 1] \quad \text{delay}(?r) & :- \text{roadwork}(?r) \wedge \text{police_act}(?r) \end{aligned}$$

The service generates advisories expressed as the likelihood of delays along the routes of interest. The students do not want to miss the show due to traffic, but they also have conference deadlines and so do not want to leave too early. They decide that if the advisory says that the likelihood of delays is between 0.2 and 0.4 then they add one extra hour to the trip time. If the likelihood is between 0.4 and 0.6, then they add two hours, and if the likelihood is over 0.6 then they take a train.

The key observation here is that the three rules used in generating the advisory are not independent—they all rely on the roadwork information from Department of Transportation. Our intuition suggests that predictions based on the independence assumption might cost our student a Broadway show, a few hours of sleep, or a conference paper.

As mentioned in the introduction, the novelty of our approach is that it does not assume that the information sources are independent and instead properly correlates inferences obtained using the rules that represent these sources. In Section 7, we will return to this example and show that our approach improves the quality of the advisory and could help the students avoid unnecessary grief.

3 Preliminaries

In BLP, uncertainty is represented using belief functions of Dempster-Shafer Theory [5, 22].

In Probability Theory, a probability distribution function assigns probabilities to mutually exclusive events. In Dempster-Shafer Theory, a mass function assigns evidence (also known as degree of belief, certainty, or support) to *sets of* mutually exclusive states. For example, the state $\{A, B, C\}$ may have an associated degree of belief 0.4, which means that either A or B or C is true with certainty 0.4. This statement does *not* imply anything about the individual truth of A , B , or C , or about any of the sets $\{A, B\}$, $\{A, C\}$, or $\{B, C\}$.

Let \mathcal{U} be the **universal set**—a set of all possible mutually exclusive states under consideration. The power set $\mathbb{P}(\mathcal{U})$ is the set of all possible sub-sets of \mathcal{U} , including the empty set, \emptyset .

A **mass function** is a mapping $\text{mass} : \mathbb{P}(\mathcal{U}) \rightarrow [0, 1]$ such that $\text{mass}(\emptyset) = 0$ and $\sum_{S \in \mathbb{P}(\mathcal{U})} \text{mass}(S) = 1$. $\text{mass}(S)$ expresses the proportion of all relevant and available evidence that supports the claim that the actual true state belongs to the set $S \subseteq \mathcal{U}$ and to no known proper subset of S . If it is also known that the actual state belongs to a subset S' of S then $\text{mass}(S')$ will also be non-zero.

The **belief** associated with the set S is defined as the sum of all the masses of S 's subsets: $\text{belief}(S) = \sum_{S' \subseteq S} \text{mass}(S')$.

Dempster's combination rule [5, 22] addresses the issue of how to combine two independent sets of mass assignments. It emphasizes the agreement between multiple sources and ignores correlation and conflict through a normalization factor. As it turns out, ignoring these aspects leads to unexpected derivations [30]. To avoid this problem, BLP supports a family of combination methods, e.g., the rules in [21, 28], and does not commit to any particular one. As a special case, Dempster's combination rule and its extensions can be used when appropriate.

4 Syntax of BLP

A **belief logic program** (or a *blp*, for short) is a set of annotated rules. Each **annotated rule** has the following format:

$$[v, w] X \text{ :- } \textit{Body}$$

where X is a positive atom and *Body* is a Boolean combination of atoms, i.e., a formula composed out of atoms by conjunction, disjunction, and negation. We will use capital letters to denote positive atoms, e.g., A , and a bar over such a letter will denote negation, e.g., \bar{A} . The annotation $[v, w]$ is called a **belief factor**, where v and w are real numbers such that $0 \leq v \leq w \leq 1$.

The informal meaning of the above rule is that if *Body* is true, then this rule supports X to the degree v and \bar{X} to the degree $1 - w$. The difference, $w - v$, is the *information gap* (or the degree of ignorance) with regard to X .

Note that, in keeping with the theory of evidence, BLP uses what is known as explicit negation (or, strong negation) [18] rather than negation as failure.

That is, if nothing is known about A , it only means that there is no evidence that A holds; it does not mean that the negation of A holds.

An annotated rule of the form $[v, w] X :- true$ is called an **annotated fact**; it is often written simply as $[v, w] X$. In the remainder of this paper we will deal only with annotated rules and facts and refer to them simply as rules and facts.

Definition 1. *Given a blp P , an atom X is said to **depend** on an atom Y*

- *directly, if X is the head of a rule R and Y occurs in the body of R ;*
- *indirectly, if X is dependent on Z , and Z depends on Y .* □

We require that in a ground blp no atom depends on itself. So, there can be no cyclic dependency among ground atoms. Most other works in this area, e.g., [3, 19, 20], make the same assumption. The extension of BLP that allows cyclic dependency is future work and is beyond the scope of this paper.

5 Semantics of BLP

We begin with the concept of *combination functions*.

5.1 Combination Functions

Definition 2. *Let D be the set of all sub-intervals of $[0, 1]$, and $\Phi : D \times D \rightarrow D$ be a function. Let us represent $\Phi([v_1, w_1], [v_2, w_2])$ as $[V(v_1, w_1, v_2, w_2), W(v_1, w_1, v_2, w_2)]$. We say that Φ is a **belief combination function** if Φ is associative and commutative.* □

A useful common-sense restriction on combination functions is that the functions V and W above are monotonically increasing in each of their four arguments, but this is not required for our results. Due to the associativity of Φ , we can extend it from two to three and more arguments as follows:

$$\Phi([v_1, w_1], \dots, [v_k, w_k]) = \Phi(\Phi([v_1, w_1], \dots, [v_{k-1}, w_{k-1}]), [v_k, w_k])$$

For convenience, we also extend Φ to the nullary case and the case of a single argument as follows: $\Phi() = [0, 1]$ and $\Phi([v, w]) = [v, w]$. Note that the order of arguments in a belief combination function is immaterial, since such functions are commutative, so we often write such functions as functions on multisets of intervals, e.g., $\Phi(\{[v_1, w_1], \dots, [v_k, w_k]\})$.

As mentioned earlier, there are many ways to combine evidence and so there are many useful belief combination functions. Different functions can be used for different application domains and even for different types of data within the same domain. Our examples will be using the following three popular functions:

- *Dempster's combination rule:*
 - $\Phi^{DS}([0, 0], [1, 1]) = [0, 1]$.
 - $\Phi^{DS}([v_1, w_1], [v_2, w_2]) = [v, w]$ if $\{[v_1, w_1], [v_2, w_2]\} \neq \{[0, 0], [1, 1]\}$, where $v = \frac{v_1 \cdot w_2 + v_2 \cdot w_1 - v_1 \cdot v_2}{K}$, $w = \frac{w_1 \cdot w_2}{K}$, and $K = 1 + v_1 \cdot w_2 + v_2 \cdot w_1 - v_1 - v_2$.
- *Maximum:* $\Phi^{max}([v_1, w_1], [v_2, w_2]) = [max(v_1, v_2), max(w_1, w_2)]$.
- *Minimum:* $\Phi^{min}([v_1, w_1], [v_2, w_2]) = [min(v_1, v_2), min(w_1, w_2)]$.

5.2 Semantics

Given a blp \mathbf{P} , the definitions of *Herbrand Universe* $U_{\mathbf{P}}$ and *Herbrand Base* $B_{\mathbf{P}}$ of \mathbf{P} are the same as in the classical case. As usual in logic programming, the easiest way to define a semantics is by considering *ground* (i.e., variable-free) rules. We assume that each atom $X \in B_{\mathbf{P}}$ has an associated belief combination function, denoted Φ_X . Intuitively, Φ_X is used to help determine the *combined* belief in X accorded by the rules in \mathbf{P} that support X .

Definition 3. A **truth valuation** over a set of atoms α is a mapping from α to $\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. The set of all possible valuations over α is denoted as $\mathcal{TV}al(\alpha)$.

A **truth valuation** I for a blp \mathbf{P} is a truth valuation over $B_{\mathbf{P}}$. Let $\mathcal{TV}al(\mathbf{P})$ denote the set of all the truth valuations for \mathbf{P} , so $\mathcal{TV}al(\mathbf{P}) = \mathcal{TV}al(B_{\mathbf{P}})$. \square

It is easy to see that $\mathcal{TV}al(\mathbf{P})$ has $3^{|B_{\mathbf{P}}|}$ truth valuations. If α is a set of atoms, we will use $\mathcal{B}ool(\alpha)$ to denote the set of all Boolean formulas constructed out of these atoms (i.e., using \wedge , \vee , and negation).

Definition 4. Given a truth valuation I over a set of atoms α and a formula $F \in \mathcal{B}ool(\alpha)$, $I(F)$ is defined as in Lukasiewicz's three-valued logic: $I(A \vee B) = \max(I(A), I(B))$, $I(A \wedge B) = \min(I(A), I(B))$, and $I(\bar{A}) = \neg I(A)$, where $\mathbf{f} < \mathbf{u} < \mathbf{t}$ and $\neg \mathbf{t} = \mathbf{f}$, $\neg \mathbf{f} = \mathbf{t}$, $\neg \mathbf{u} = \mathbf{u}$. We say that $I \models F$ if $I(F) = \mathbf{t}$. \square

Definition 5. A **support function** for a set of atoms α is a mapping m_{α} from $\mathcal{TV}al(\alpha)$ to $[0, 1]$ such that $\sum_{I \in \mathcal{TV}al(\alpha)} m_{\alpha}(I) = 1$.

The atom-set α is called the **base** of m_{α} . A **support function** for a blp \mathbf{P} is a mapping m from $\mathcal{TV}al(\mathbf{P})$ to $[0, 1]$ such that $\sum_{I \in \mathcal{TV}al(\mathbf{P})} m(I) = 1$. \square

Support functions, defined above, are always associated with mass functions of Dempster-Shafer theory, as discussed in Section 6. In Dempster-Shafer theory, every mass function has a corresponding belief function and, similarly, BLP support functions are associated with belief functions, defined next.

Definition 6. Recall that $\mathcal{B}ool(B_{\mathbf{P}})$ denotes the set of all Boolean formulas composed out of the atoms in $B_{\mathbf{P}}$. A mapping $\mathbf{bel} : \mathcal{B}ool(B_{\mathbf{P}}) \rightarrow [0, 1]$ is said to be a **belief function** for \mathbf{P} if there exists a support function m for \mathbf{P} , so that for all $F \in \mathcal{B}ool(B_{\mathbf{P}})$

$$\mathbf{bel}(F) = \sum_{I \in \mathcal{TV}al(\mathbf{P}) \text{ such that } I \models F} m(I) \quad \square$$

Belief functions can be thought of as *interpretations* of belief logic programs. However, as usual in deductive database and logic programming, we are interested not just in interpretations, but in models. We define this next.

Definition 7. Given a blp \mathbf{P} and a truth valuation I , we define \mathbf{P} 's **reduct under** I to be $\mathbf{P}_I = \{R \mid R \in \mathbf{P}, I \models \text{Body}(R)\}$, where $\text{Body}(R)$ denotes the body of the rule R .

Let $\mathbf{P}(X)$ denote the set of rules in \mathbf{P} with the atom X in the head. \mathbf{P} 's **reduct under** I **with** X **as head** is defined as $\mathbf{P}_I(X) = \mathbf{P}_I \cap \mathbf{P}(X)$. Thus, $\mathbf{P}_I(X)$ is simply that part of the reduct \mathbf{P}_I , which consists of the rules that have X as their head. \square

We now define a measure for the degree by which I is supported by $\mathbf{P}(X)$.

Definition 8. Given a blp \mathbf{P} and a truth valuation I for \mathbf{P} , for any $X \in B_{\mathbf{P}}$, we define $s_{\mathbf{P}}(I, X)$, called the **\mathbf{P} -support for X in I** , as follows.

1. If $\mathbf{P}_I(X) = \phi$, then
 - If $I(X) = \mathbf{t}$ or $I(X) = \mathbf{f}$, then $s_{\mathbf{P}}(I, X) = 0$;
 - If $I(X) = \mathbf{u}$, then $s_{\mathbf{P}}(I, X) = 1$.
2. If $\mathbf{P}_I(X) = \{R_1, \dots, R_n\}$, $n > 0$, let $[v, w]$ be the result of applying Φ_X to the belief factors of the rules R_1, \dots, R_n . Then
 - If $I(X) = \mathbf{t}$, then $s_{\mathbf{P}}(I, X) = v$;
 - If $I(X) = \mathbf{f}$, then $s_{\mathbf{P}}(I, X) = 1 - w$;
 - If $I(X) = \mathbf{u}$, then $s_{\mathbf{P}}(I, X) = w - v$. □

Informally, $I(X)$ represents what the possible world I believes about X . The above interval $[v, w]$ produced by the Φ_X represents the combined support accorded by the rule set $\mathbf{P}_I(X)$ to that belief. $s_{\mathbf{P}}(I, X)$ measures the degree by which a truth valuation I is supported by $\mathbf{P}(X)$. If X is true in I , it is the combined belief in X supported by \mathbf{P} given the truth valuation I . If X is false in I , $s_{\mathbf{P}}(I, X)$ is the combined disbelief in X . Otherwise, it represents the combined information gap about X .

It is easy to see that the case of $\mathbf{P}_I(X) = \emptyset$ in the above definition is just a special case of $\mathbf{P}_I(X) = \{R_1, \dots, R_n\}$, since $\Phi_X(\emptyset)$ is $[0, 1]$, by Definition 2.

We now introduce the notion of \mathbf{P} -support for I as a whole. It is defined as a cumulative \mathbf{P} -support for all atoms in the Herbrand base.

Definition 9. If I is a truth valuation for a blp \mathbf{P} , then

$$\hat{m}_{\mathbf{P}}(I) = \prod_{X \in B_{\mathbf{P}}} s_{\mathbf{P}}(I, X) \quad \square$$

Theorem 1. For any blp \mathbf{P} , $\sum_{I \in \mathcal{TVal}(\mathbf{P})} \hat{m}_{\mathbf{P}}(I) = 1$. □

In other words, $\hat{m}_{\mathbf{P}}$ is a support function. This theorem is crucial, as it makes the following definition well-founded.

Definition 10. The **model** of a blp \mathbf{P} is the following belief function:

$$\text{model}(F) = \sum_{I \in \mathcal{TVal}(\mathbf{P}) \text{ such that } I \models F} \hat{m}_{\mathbf{P}}(I), \quad \text{where } F \in \mathcal{Bool}(B_{\mathbf{P}}). \quad \square$$

The belief function $\text{model}(F)$ measures the degree by which F is supported by \mathbf{P} . It is easy to see that every blp has a *unique* model. The rationale for the above definition is expressed by the following theorem:

Theorem 2. Let \mathbf{P} be a blp and A an atom. For any rule R , let $\text{Body}(R)$ denote its body. Let \mathcal{S} be a subset of $\mathbf{P}(A)$ that satisfies (i) $\text{model}(\bigwedge_{R \in \mathcal{S}} \text{Body}(R)) > 0$; and (ii) \mathcal{S} is maximal: if $\mathcal{S}' \supseteq \mathcal{S}$ is another subset of $\mathbf{P}(A)$ that satisfies (i) then $\mathcal{S}' = \mathcal{S}$. Let $[v_R, w_R]$ denote the belief factor associated with the rule R

and suppose $\Phi_A(\{[v_R, w_R]\}_{R \in \mathcal{S}}) = [v, w]$ ($\Phi_A(\{[v_R, w_R]\}_{R \in \mathcal{S}}$) is the result of applying Φ_A to the belief factors in \mathcal{S}). Then

$$\frac{\mathbf{model}(A \wedge \bigwedge_{R \in \mathcal{S}} \mathit{Body}(R))}{\mathbf{model}(\bigwedge_{R \in \mathcal{S}} \mathit{Body}(R))} = v \qquad \frac{\mathbf{model}(\overline{A} \wedge \bigwedge_{R \in \mathcal{S}} \mathit{Body}(R))}{\mathbf{model}(\bigwedge_{R \in \mathcal{S}} \mathit{Body}(R))} = 1 - w. \quad \square$$

In other words, \mathbf{model} is a “correct” (and unique) belief function that embodies the evidence that \mathbf{P} provides for each atom. In other words, \mathbf{model} supports each atom in the Herbrand base with precisely the expected amount of support. In contrast, all other works that we are aware of either do not account for combined support provided by multiple rules deriving the same atom or do not have a clear model-theoretic account for that phenomenon.

It is not hard to see that the BLP semantics is non-monotonic.² To see that, suppose rule r_1 has the form $[0.4, 0.4] X$ and rule r_2 is $[0.8, 0.8] X$. Let \mathbf{P}_1 be $\{r_1\}$, \mathbf{P}_2 be $\{r_2\}$, \mathbf{P}_3 be $\{r_1, r_2\}$, and \mathbf{bel}_i be the model of \mathbf{P}_i , $i = 1, 2, 3$. For any combination function Φ , let $[v, w] = \Phi([0.4, 0.4], [0.8, 0.8])$, since $v \leq w$, either $v < 0.8$ is true, or $w > 0.4$ is true, or both. If $v < 0.8$, then $\mathbf{bel}_3(X) < 0.8 = \mathbf{bel}_2(X)$. Thus, adding r_1 to \mathbf{P}_2 reduces the support for X . If $w > 0.4$, then $\mathbf{bel}_3(\overline{X}) < 0.6 = \mathbf{bel}_1(\overline{X})$, meaning that adding r_2 to \mathbf{P}_1 reduces the support for \overline{X} . Non-monotonicity of Dempster-Shafer theory was also discussed in [16].

Also, under the BLP semantics, the support for A provided by a rule of the form $[v, w] A :- B_1 \vee B_2$ might differ from the support for A provided by the pair of rules $[v, w] A :- B_1$ and $[v, w] A :- B_2$, if $\Phi_A([v, w], [v, w]) \neq [v, w]$.

A direct implementation of the semantics would have high complexity. A much more efficient query answering algorithm is presented in [26].

6 Discussion

First, one might be wondering whether the combination functions in BLP are really necessary and whether the same result could not be achieved without the use of combination functions. The answer is that combination functions *can* be dispensed with. However, this requires an extension of BLP with default negation and, more importantly, causes an exponential blowup of the program (making it an unlikely tool for knowledge engineering). This theme will be elaborated upon in a full version of this paper.

Next we discuss the relationship of BLP to Dempster-Shafer belief functions and defeasible reasoning.

Probability vs. belief in combination of evidence. Probability theory has been widely used for reasoning with uncertainty. However, several aspects of the

² However, monotonicity holds under certain conditions, for instance, if every belief factor is of the form $[v, 1]$ and every combination function $\Phi([v_1, 1], [v_2, 1])$ is monotonically increasing in v_1 and in v_2 . Under these conditions, the belief in any negated literal is 0.

application of this theory to modeling uncertainty has been criticized [22, 31], especially when it comes to combining evidence obtained from different sources.

To illustrate, consider two mutually exclusive states A and \bar{A} . Suppose this distribution is provided by two different sources. Source 1 may assert that $prob(A) = 0.8$, $prob(\bar{A}) = 0.2$, meaning that the probability of A is 0.8. Source 2 may assert that $prob(A) = 0.6$, $prob(\bar{A}) = 0.4$, meaning that the probability of A is 0.6. There is no obvious way to combine information from these two sources because probability is objective. Some approaches take the maximum or the minimum of the two probability values; others take the average. However, none of these has any probabilistic justification.

In some frameworks, e.g. [4, 15], probability intervals are used to model uncertainty. Suppose source 1 asserts $0.8 \leq prob(A) \leq 1$ and source 2 asserts $0.6 \leq prob(A) \leq 0.7$. Some approaches [4] compute the intersection of the two intervals, yielding \emptyset (thus concluding nothing). Some other approaches [15] simply combine the uncertainty ranges, for instance, $[min(0.8, 0.6), max(1, 0.7)]$. Again, no probabilistic justification exists for either of these rules of combination, so probability theory is used here in name but not in substance.

In contrast, Dempster-Shafer theory [5, 22] gives up certain postulates of the probability theory in order to provide an account for the phenomenon of combined evidence.

Relationship to Dempster-Shafer theory of evidence. We now relate our semantics to Dempster-Shafer theory.

Definition 11. A *complete valuation* over a set of atoms α is a mapping from α to $\{\mathbf{t}, \mathbf{f}\}$. The set of all complete valuations over α is denoted as $\mathcal{U}(\alpha)$. A *complete valuation* I for a blp \mathbf{P} is a complete valuation over $B_{\mathbf{P}}$. Let $\mathcal{U}(\mathbf{P})$ denote the set of all the complete valuations for \mathbf{P} , so $\mathcal{U}(\mathbf{P}) = \mathcal{U}(B_{\mathbf{P}})$. \square

Complete valuations correspond to interpretations in classical 2-values logic programs. A complete valuation J can be viewed as a state, and it is clear that two different complete valuations represent two mutually exclusive states. $\mathcal{U}(\alpha)$ is a universal set of mutually exclusive states.

A truth valuation I over α (see Definition 3) can be uniquely mapped to a set of complete valuations: $\Psi_I = \{J \in \mathcal{U}(\alpha) \mid \forall A \in \alpha, J(A) = \mathbf{t} \text{ if } I(A) = \mathbf{t}, J(A) = \mathbf{f} \text{ if } I(A) = \mathbf{f}\}$. In other words, each truth valuation I represents a subset Ψ_I of $\mathcal{U}(\alpha)$, and hence an element Ψ_I of $\mathbb{P}(\mathcal{U}(\alpha))$, the power set of $\mathcal{U}(\alpha)$. Obviously, $\{\Psi_I \mid I \in \mathcal{TV}al(\alpha)\} \subset \mathbb{P}(\mathcal{U}(\alpha))$.

Dempster-Shafer's mass function, **mass**, is a mapping from $\mathbb{P}(\mathcal{U}(\alpha))$ to $[0, 1]$ such that $\sum_{S \in \mathbb{P}(\mathcal{U}(\alpha))} \mathbf{mass}(S) = 1$. According to Definition 5, our support function m is a mapping from $\mathcal{TV}al(\alpha)$ to $[0, 1]$ such that $\sum_{I \in \mathcal{TV}al(\alpha)} m(I) = 1$. Given any support function m , it can be related to a unique mass function, **mass**, as follows.

$$\begin{aligned} \mathbf{mass}(\Psi_I) &= m(I), & \forall I \in \mathcal{TV}al(\alpha) \\ \mathbf{mass}(S) &= 0, & \text{if } S \neq \Psi_I, \forall I \in \mathcal{TV}al(\alpha) \end{aligned} \quad (1)$$

Based on the above correspondence between Dempster-Shafer's mass function and BLP support function introduced in Definition 5, we establish a correspondence between Dempster-Shafer's belief function and BLP's belief function of Definition 6.

Any BLP formula $F \in \mathcal{Bool}(B_{\mathcal{P}})$ (defined in Section 5.2) uniquely corresponds to a set of complete valuations: $\Theta_F = \{J \in \mathcal{U}(B_{\mathcal{P}}) \mid J \models F\}$. Clearly, $\Theta_F \in \mathbb{P}(\mathcal{U}(B_{\mathcal{P}}))$, and it can be shown that $\{\Theta_F \mid F \in \mathcal{Bool}(B_{\mathcal{P}})\} = \mathbb{P}(\mathcal{U}(B_{\mathcal{P}}))$.

Theorem 3. *Let m be a support function and \mathbf{mass} its corresponding mass function as in (1). Also let \mathbf{bel} be the belief function of Definition 6 constructed from m , and let \mathbf{belief} be the Dempster-Shafer's belief function based on \mathbf{mass} : $\mathbf{belief}(S) = \sum_{\forall S' \subseteq S} \mathbf{mass}(S')$. Then, for any $F \in \mathcal{Bool}(B_{\mathcal{P}})$, the following holds: $\mathbf{bel}(F) = \mathbf{belief}(\Theta_F)$. \square*

In other words, any BLP belief function corresponds to a Dempster-Shafer's belief function.

Relationship to defeasible logic programs and explicit negation. There is an interesting correspondence between the treatment of contradictory information in BLP and a form of defeasible reasoning called Courteous Logic Programming [7] and, more generally, Logic Programming with Courteous Argumentation Theories (LPDA) [25]. Here we consider only LPDA without default negation. A defeasible LPDA rule has the form

$$\textcircled{r} H \text{ :- } B_1 \wedge \dots \wedge B_n \quad (2)$$

where r is a label, H and B_i , $1 \leq i \leq n$, are atoms or explicitly negated atoms. As before, we use \bar{A} to represent explicit negation of A .

For any atom A , let $\lambda(A) = [1, 1] A$ and $\lambda(\bar{A}) = [0, 0] A$. We extend λ to rules of the form (2) so that $\lambda(\textcircled{r} H \text{ :- } B_1 \wedge \dots \wedge B_n)$ is $\lambda(H) \text{ :- } B_1 \wedge \dots \wedge B_n$. Finally, λ is extended to programs so that $\lambda(\Pi) = \{\lambda(R) \mid R \in \Pi\}$. Note that $\lambda(\Pi)$ is a blp.

Let $\Pi \models_{LPDA} F$ denote that Π entails F under the semantics of LPDA [25] with one of the courteous argumentation theories and let the combination function Φ_1 be such that $\Phi_1([0, 0], [1, 1]) = [0, 1]$, $\Phi_1([0, 0], [0, 0]) = [0, 0]$, $\Phi_1([1, 1], [1, 1]) = [1, 1]$.

Theorem 4. *Let Π be an acyclic LPDA program that consists of the rules of the form (2) and let $\mathbf{bel}_{\lambda(\Pi)}$ be the model of the blp $\lambda(\Pi)$ with the combination function Φ_1 . Assume, in addition, that none of the rules in Π defines or uses the special predicates *overrides* and *opposes*, which in LPDA provide information about conflicting rules and their defeasibility properties. Then, for any formula $F \in \mathcal{Bool}(B_{\Pi})$, $\Pi \models_{LPDA} F$ if and only if $\mathbf{bel}_{\lambda(\Pi)}(F) = 1$. \square*

In both theories, the presence of A and \bar{A} (in LPDA) or of $[1, 1] A$ and $[0, 0] A$ (in BLP) implies that A 's truth value is undefined. That is, inconsistent information is self-defeating. In contrast, Pearce and Wagner's logic programs with strong negation, as defined in [18], handle inconsistent information by explicitly declaring a contradiction. Thus, if Π in Theorem 4 is a program with strong

negation then $\Pi \models_{LPSN} F$ is equivalent to $\text{bel}_{\lambda(\Pi)}(F) = 1$ only if Π is *consistent*. Here \models_{LPSN} denotes the entailment in logic programming with strong negation [18].

In the opposite direction, the connection is more complicated and we do not have the space to describe it here.

7 Motivating Example (cont'd)

Returning to the example in Section 2, suppose that our information sources predict 50% chance of bad weather, parades with 50% certainty, roadwork along the Long Island Expressway (henceforth LIE) with certainty 80%, and police activity due to accidents with the likelihood of 40%. This information is expressed in BLP as follows.

$$\begin{array}{ll} [0.8, 0.8] \text{ roadwork}(LIE) & [0.5, 0.5] \text{ bad_weather}(LIE) \\ [0.5, 0.5] \text{ social_act}(LIE) & [0.4, 0.4] \text{ police_act}(LIE) \end{array} \quad (3)$$

The traffic service fetches the above information from four different information sources and integrates them using these rules: ³

$$\begin{array}{l} [0.9, 1] \text{ delay}(?r) \text{ :- roadwork}(?r) \wedge \text{bad_weather}(?r) \\ [0.8, 1] \text{ delay}(?r) \text{ :- roadwork}(?r) \wedge \text{social_act}(?r) \\ [0.99, 1] \text{ delay}(?r) \text{ :- roadwork}(?r) \wedge \text{police_act}(?r) \end{array}$$

Suppose the atom $\text{delay}(?r)$ is associated with the combination function Φ^{max} defined in Section 5.1. When correlation is not taken into account, as in [1, 12], the belief factor of $\text{delay}(LIE)$ is $[0.36, 1]$, which means that the available information predicts traffic delay with certainty 0.36 and smooth traffic with certainty 0. Based on this advisory, the students would decide to drive and leave one hour earlier than normal (see Section 2 for the explanation of how the students make their travel plans). It is not hard to see that this advisory might cost our students a show. The information from the weather forecast and Department of Transportation alone (the first rule) is enough to predict traffic delays with certainty 0.36. Taking into account the possibilities of parades and accidents, it is reasonable to up the expectation of delays. In contrast, BLP computes the belief factor for traffic delays to be $[0.63, 1]$, which means that our students will shell out a little extra for the train but will make it to the show.

One may argue that the problem was the Φ^{max} combination function and a different function, such as the Dempster's combination rule Φ^{DS} (Section 5.1), may do just fine even without BLP. While this might be true for the traffic conditions in (3), a wrong advice will be given in the following scenario:

$$\begin{array}{ll} [0.2, 0.2] \text{ roadwork}(LIE) & [0.8, 0.8] \text{ bad_weather}(LIE) \\ [0.9, 0.9] \text{ social_act}(LIE) & [0.3, 0.3] \text{ police_act}(LIE) \end{array}$$

³ Note that although the semantics is defined for ground rules, query answering algorithms work with non-ground rules [26].

where we assume that $delay(?r)$ is associated with the combination function Φ^{DS} . Without taking correlation into account, the belief factor of $delay(LIE)$ becomes $[0.31, 1]$ —again suggesting to add one extra hour to the trip. However, this advisory errs on the cautious side. Here all three rules make their predictions building the same roadwork factor into their decision, so this factor is counted multiple times. In contrast, BLP recognizes that the three predictions are not independent and its calculated certainty factor is $[0.18, 1]$. Thus, the students will allocate no extra time for the eventualities and will get that badly needed extra hour of sleep before their conference deadlines.

We thus see that, by correlating rules, BLP is able to better predict certainty factors of the combined information.

8 Conclusions

We introduced a novel logic theory, Belief Logic Programming, for reasoning with uncertainty. BLP is based on the concept of belief function and is inspired by Dempster-Shafer theory, but it is not simply integration of Dempster-Shafer theory and logic programming. First, unlike the previous efforts in applying Dempster-Shafer theory in logic programming [1, 2, 15], BLP can correlate structural information contained in derivation paths for beliefs, as illustrated in the motivating example in Section 2 and Section 7. Secondly, BLP is not restricted to any particular combination rule, instead any number of reasonable combination rules can be used. Apart from traditional uses in expert systems, such a language can be used to integrate semantic Web services and information sources, such as sensor networks and forecasts, which deal with uncertain data.

For future work, we plan to extend the algorithms to deal with non-ground rules and queries, and to make them optimized based on belief factors given in the query. Another important extension is to allow cyclic dependency among ground atoms.

References

1. J. F. Baldwin. Support logic programming. *Intl. Journal of Intelligent Systems*, 1:73–104, 1986.
2. J. F. Baldwin. Evidential support logic programming. *Fuzzy Sets and Systems*, 24(1):1–26, 1987.
3. U. Bergsten and J. Schubert. Dempster’s rule for evidence ordered in a complete directed acyclic graph. *Int. J. Approx. Reasoning*, 9:37–73, 1993.
4. A. Dekhtyar and V. S. Subrahmanian. Hybrid probabilistic programs. *J. of Logic Programming*, 43:391–405, 1997.
5. A. P. Dempster. Upper and lower probabilities induced by a multi-valued mapping. *Ann. Mathematical Statistics*, 38, 1967.
6. C. Elkan. The paradoxical success of fuzzy logic. In *IEEE Expert*, pages 698–703, 1993.
7. B.N. Grosz. A courteous compiler from generalized courteous logic programs to ordinary logic programs. Technical Report Supplementary Update Follow-On to RC 21472, IBM, July 1999.

8. J. Y. Halpern. *Reasoning About Uncertainty*. MIT Press, 2003.
9. K. Kersting and L. De Raedt. Bayesian logic programs. Technical report, Albert-Ludwigs University at Freiburg, 2001.
10. M. Kifer and A. Li. On the semantics of rule-based expert systems with uncertainty. In *ICDT '88: Intl. Conf. on Database Theory*, pages 102–117, London, UK, 1988. Springer-Verlag.
11. M. Kifer and V. S. Subrahmanian. Theory of generalized annotated logic programming and its applications. *J. of Logic Programming*, 12(3,4):335–367, 1992.
12. L. V. S. Lakshmanan and N. Shiri. A parametric approach to deductive databases with uncertainty. *IEEE Trans. on Knowledge and Data Engineering*, 13(4):554–570, 2001.
13. T. Lukasiewicz. Probabilistic logic programming with conditional constraints. *ACM Trans. on Computational Logic*, 2(3):289–339, 2001.
14. S. Muggleton. Learning stochastic logic programs. *Electron. Trans. Artif. Intell.*, 4(B):141–153, 2000.
15. R. T. Ng. Reasoning with uncertainty in deductive databases and logic programs. *Intl. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 5(3):261–316, 1997.
16. R. T. Ng and V. S. Subrahmanian. Relating dempster-shafer theory to stable semantics. In *ISLP*, pages 551–565, 1991.
17. R. T. Ng and V. S. Subrahmanian. A semantical framework for supporting subjective probabilities in deductive databases. *J. of Automated Reasoning*, 10(2):191–235, 1993.
18. D. Pearce and G. Wagner. Logic programming with strong negation. In *Proceedings of the international workshop on Extensions of logic programming*, pages 311–326, New York, NY, USA, 1991. Springer-Verlag New York, Inc.
19. D. Poole. The independent choice logic and beyond. In *Probabilistic Inductive Logic Programming*, pages 222–243, 2008.
20. L. De Raedt and K. Kersting. Probabilistic inductive logic programming. In *Probabilistic Inductive Logic Programming*, pages 1–27, 2008.
21. I. Ruthven and M. Lalmas. Using dempster-shafers theory of evidence to combine aspects of information use. *J. of Intelligent Systems*, 19:267–301, 2002.
22. G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
23. P. P. Shenoy and G. Shafer. Axioms for probability and belief-function propagation. In *Uncertainty in Artificial Intelligence*, pages 169–198. North-Holland, 1990.
24. V. S. Subrahmanian. On the semantics of quantitative logic programs. In *SLP*, pages 173–182, 1987.
25. H. Wan, B.N. Grosf, M. Kifer, P. Fodor, and S. Liang. Logic programming with defaults and argumentation theories. In *Intl. Conf. on Logic Programming*, 2009.
26. H. Wan and M. Kifer. Query answering in belief logic programming. In *Intl. Conf. on Scalable Uncertainty Management (SUM)*, 2009.
27. R. R. Yager. Decision making under dempster-shafer uncertainties. In *Classic Works on the Dempster-Shafer Theory of Belief Functions*, pages 619–632. Springer, 2008.
28. K. Yamada. A new combination of evidence based on compromise. *Fuzzy Sets System*, 159(13):1689–1708, 2008.
29. L. A. Zadeh. Fuzzy sets. *Information Control*, 8:338–353, 1965.
30. L. A. Zadeh. A review of (a mathematical theory of evidence. g. shafer, princeton university press, princeton, nj, 1976). *The AI Magazine*, 1984.
31. L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets Syst.*, 100(supp.):9–34, 1999.