Markov Logic Style Weighted Rules under the Stable Model Semantics

JOOHYUNG LEE
Arizona State University, Tempe, AZ, USA (*e-mail:* `joolee@asu.edu`)

YUNSONG MENG
Samsung Research America, Mountain View, CA, USA (*e-mail:* `yunsong.m@samsung.com`)

YI WANG
Arizona State University, Tempe, USA (*e-mail:* `ywang485@asu.edu`)

submitted 29 April 2015; accepted 5 June 2015

Abstract

We introduce the language LP$^{\rm MLN}$ that extends logic programs under the stable model semantics to allow weighted rules similar to the way Markov Logic considers weighted formulas. LP$^{\rm MLN}$ is a proper extension of the stable model semantics to enable probabilistic reasoning, providing a way to handle inconsistency in answer set programming. We also show that the recently established logical relationship between Pearl's Causal Models and answer set programs can be extended to the probabilistic setting via LP$^{\rm MLN}$.

KEYWORDS: Answer Set Programming, Markov Logic Networks, Probabilistic Causal Models, Probabilistic Logic Programming

1 Introduction

Logic programs under the stable model semantics is the language for Answer Set Programming (ASP). Many useful knowledge representation constructs are introduced into ASP, and several efficient ASP solvers are available. However, like many "crisp" logic approaches, adding a single rule may easily introduce inconsistency. Also, ASP is not well suited for handling probabilistic reasoning.

Markov Logic is a successful approach to combining first-order logic and probabilistic graphical models in a single representation. Syntactically, a Markov Logic Network (MLN) is a set of weighted first-order logic formulas. Semantically, the probability of each possible world is derived from the sum of the weights of the formulas that are true under the possible world. Markov Logic has shown to formally subsume many other SRL languages and has been successfully applied to several challenging applications, such as natural language processing and entity resolution. However, the logical component of Markov Logic is the standard first-order logic semantics, which does not handle the concept of rules as in ASP.

We introduce a simple approach to combining the two successful formalisms, which allows for logical reasoning originating from the stable model semantics as well as probabilistic reasoning originating from Markov Logic. LP$^{\rm MLN}$ is a proper extension of the standard stable model semantics, and as such embraces the rich body of research in answer set programming.

Interestingly, the relationship between LP$^{\rm MLN}$ and Markov Logic is analogous to the well-known relationship between ASP and SAT (Lin and Zhao 2004; Lee 2005). This allows many useful results known in the deterministic case to be carried over to the probabilistic setting. In particular, an implementation of Markov Logic can be used to compute "tight" LP$^{\rm MLN}$ programs, similar to the way "tight" ASP programs can be computed by SAT solvers.

LP$^{\rm MLN}$ provides a viable solution to inconsistency handling with ASP knowledge bases. For example, consider ASP knowledge base KB_1 that states that *Man* and *Woman* are disjoint subclasses of *Human*.

$$\begin{array}{rcl} \mathit{Human}(x) & \leftarrow & \mathit{Man}(x), \\ \mathit{Human}(x) & \leftarrow & \mathit{Woman}(x), \\ & \leftarrow & \mathit{Man}(x), \mathit{Woman}(x) \end{array}$$

One data source KB_2 says that *Jo* is a *Man*:

$$\mathit{Man}(\mathit{Jo})$$

while another data source KB_3 states that *Jo* is a *Woman*:

$$\mathit{Woman}(\mathit{Jo}).$$

The data about *Jo* is actually inconsistent, so under the (deterministic) stable model semantics, the combined knowledge base $KB = KB_1 \cup KB_2 \cup KB_3$ is inconsistent, and may derive any conclusion. On the other hand, it is intuitive to view that one of the data sources may be wrong, and we still want to conclude that *Jo* is a *Human*. The same conclusion is obtained under the LP$^{\rm MLN}$ semantics.

For another aspect of LP$^{\rm MLN}$, we consider how it is related to Pearl's Probabilistic Causal Models. Both answer set programs and Probabilistic Causal Models allow for representing causality, but a precise relationship between them is established only in a recent paper (Bochman and Lifschitz 2015) limited to the deterministic case.[1] Generalizing this result to the probabilistic case is straightforward once we refer to LP$^{\rm MLN}$ in place of answer set programs, which illustrates that LP$^{\rm MLN}$ is a natural probabilistic extension of answer set programs.

2 Preliminaries

Throughout this paper, we assume a first-order signature σ that contains no function constants of positive arity. There are finitely many Herbrand interpretations of σ.

2.1 Review: Stable Model Semantics

A *rule* over σ is of the form

$$A_1; \dots ; A_k \ \leftarrow \ A_{k+1}, \dots , A_m, \mathit{not}\ A_{m+1}, \dots , \mathit{not}\ A_n, \mathit{not}\ \mathit{not}\ A_{n+1}, \dots , \mathit{not}\ \mathit{not}\ A_p \tag{1}$$

($0 \le k \le m \le n \le p$) where all A_i are atoms of σ possibly containing object variables. We write $\{A_1\}^{\rm ch} \leftarrow \mathit{Body}$ to denote the rule $A_1 \leftarrow \mathit{Body}, \mathit{not}\ \mathit{not}\ A_1$. This expression is called a "choice rule" in ASP.

[1] Strictly speaking, the relationship shown in that paper is between Pearl's causal models and nonmonotonic causal theories (McCain and Turner 1997). The close relationship between nonmonotonic causal theories and answer set programs is shown in (Ferraris et al. 2012).

We will often identify (1) with the implication:

$$A_1 \vee \cdots \vee A_k \;\leftarrow\; A_{k+1} \wedge \ldots \wedge A_m \wedge \neg A_{m+1} \wedge \ldots \wedge \neg A_n \wedge \neg\neg A_{n+1} \wedge \ldots \wedge \neg\neg A_p\ . \quad (2)$$

A *logic program* is a finite set of rules. A logic program is called *ground* if it contains no variables.

We say that an Herbrand interpretation I is a *model* of a ground program Π if I satisfies all implications (2) in Π (as in classical logic). Such models can be divided into two groups: "stable" and "non-stable" models, which are distinguished as follows. The *reduct* of Π relative to I, denoted Π^I, consists of "$A_1 \vee \cdots \vee A_k \;\leftarrow\; A_{k+1} \wedge \cdots \wedge A_m$" for all rules (2) in Π such that $I \models \neg A_{m+1} \wedge \cdots \wedge \neg A_n \wedge \neg\neg A_{n+1} \wedge \cdots \wedge \neg\neg A_p$. The Herbrand interpretation I is called a *(deterministic) stable model* of Π if I is a minimal Herbrand model of Π^I. (Minimality is in terms of set inclusion. We identify an Herbrand interpretation with the set of atoms that are true in it.)

The definition is extended to any non-ground program Π by identifying it with $gr_\sigma[\Pi]$, the ground program obtained from Π by replacing every variable with every ground term of σ.

The semantics was extended in many ways, e.g., allowing some useful constructs, such as aggregates and abstract constraints (e.g., (Niemelä and Simons 2000; Faber et al. 2004; Ferraris 2005; Son et al. 2006; Pelov et al. 2007)). The probabilistic extension defined in this paper is orthogonal to such extensions and can easily incorporate them as well.

3 Language LP$^{\rm MLN}$

3.1 Syntax of LP$^{\rm MLN}$

The syntax of LP$^{\rm MLN}$ defines a set of weighted rules. More precisely, an LP$^{\rm MLN}$ program $\mathbb{P}$ is a finite set of weighted rules $w : R$, where R is a rule of the form (1) and w is either a real number or the symbol α denoting the "infinite weight." We call rule $w : R$ *soft* rule if w is a real number, and *hard* rule if w is α.

We say that an LP$^{\rm MLN}$ program is *ground* if its rules contain no variables. We identify any LP$^{\rm MLN}$ program $\mathbb{P}$ of signature σ with a ground LP$^{\rm MLN}$ program $gr_\sigma[\mathbb{P}]$, whose rules are obtained from the rules of $\mathbb{P}$ by replacing every variable with every ground term of σ. The weight of a ground rule in $gr_\sigma[\mathbb{P}]$ is the same as the weight of the rule in $\mathbb{P}$ from which the ground rule is obtained.

By $\overline{\mathbb{P}}$ we denote the logic program obtained from $\mathbb{P}$ by dropping the weights, i.e., $\overline{\mathbb{P}} = \{R \mid w : R \in \mathbb{P}\}$. By $\overline{\mathbb{P}}_I$ we denote the set of rules in $\overline{\mathbb{P}}$ which are satisfied by I.

3.2 Semantics of LP$^{\rm MLN}$

A model of an MLN does not have to satisfy all formulas in the MLN. For each model, there is a unique maximal subset of the formulas that are satisfied by the model, and the weights of the formulas in that subset determine the probability of the model.

Likewise, a stable model of an LP$^{\rm MLN}$ program does not have to be obtained from the whole program. Instead, each stable model is obtained from some subset of the program, and the weights of the rules in that subset determine the probability of the stable model. At first, it may not seem obvious if there is a *unique* maximal subset that derives such a stable model. Nevertheless, it follows from the following proposition that this is indeed the case, and that the subset is exactly $\overline{\mathbb{P}}_I$.

Proposition 1
For any logic program Π and any subset Π' of Π, if I is a (deterministic) stable model of Π' and I satisfies Π, then I is a (deterministic) stable model of Π as well.

The proposition tells us that if I is a stable model of a program, adding additional rules to this program does not affect that I is a stable model of the resulting program as long as I satisfies the rules added. On the other hand, it is clear that I is no longer a stable model if I does not satisfy at least one of the rules added.

Thus we define the *weight* of an interpretation I w.r.t. $\mathbb{P}$, denoted $W_{\mathbb{P}}(I)$, as

$$W_{\mathbb{P}}(I) = exp\Bigg(\sum_{\substack{w:R\ \in\ \mathbb{P} \\ I\models R}} w\Bigg).$$

Let SM[$\mathbb{P}$] be the set $\{I \mid I \text{ is a stable model of } \overline{\mathbb{P}}_I\}$. Notice that SM[$\mathbb{P}$] is never empty because it always contains the empty set. It is easy to check that the set $\emptyset$ always satisfies $\overline{P}_\emptyset$, and it is the smallest set that satisfies the reduct $(\overline{P}_\emptyset)^\emptyset$.

Using this notion of a weight, we define the *probability* of an interpretation I under $\mathbb{P}$, denoted $Pr_{\mathbb{P}}[I]$, as follows. For any interpretation I,

$$Pr_{\mathbb{P}}[I] = \begin{cases} \lim\limits_{\alpha\to\infty} \frac{W_{\mathbb{P}}(I)}{\sum\limits_{J\in \text{SM}[\mathbb{P}]} W_{\mathbb{P}}(J)} & \text{if } I \in \text{SM}[\mathbb{P}]; \\ 0 & \text{otherwise.} \end{cases}$$

We omit the subscript $\mathbb{P}$ if the context is clear. We say that I is a *(probabilistic) stable model* of $\mathbb{P}$ if $Pr_{\mathbb{P}}[I] \neq 0$.

The intuition here is similar to that of Markov Logic. For each interpretation I, we try to find a maximal subset (possibly empty) of $\overline{\mathbb{P}}$ for which I is a stable model (under the standard stable model semantics). In other words, the LP$^{\text{MLN}}$ semantics is similar to the MLN semantics except that the possible worlds are the *stable* models of some maximal subset of $\overline{\mathbb{P}}$, and the probability distribution is over these stable models.

For any proposition A, $Pr_{\mathbb{P}}[A]$ is defined as:

$$Pr_{\mathbb{P}}[A] = \sum_{I:\ I\models A} Pr_{\mathbb{P}}[I].$$

(In place of "$I \models A$," one might expect "I is a stable model of $\mathbb{P}$ that satisfies A." The change does not affect the definition.)

Conditional probability under $\mathbb{P}$ is defined as usual. For propositions A and B,

$$Pr_{\mathbb{P}}[A \mid B] = \frac{Pr_{\mathbb{P}}[A \wedge B]}{Pr_{\mathbb{P}}[B]}.$$

The following example illustrates how inconsistency can be handled in LP$^{\text{MLN}}$.

Example 1 (*handling inconsistency*)
Consider the example in Section 1. Recall that there are no deterministic stable models of *KB*. However, when we identify each rule as a hard rule under the LP$^{\text{MLN}}$ semantics (i.e., having α as the weight), there are 3 probabilistic stable models (with non-zero probabilities) assuming that *Jo* is the only element in the domain. Let $Z = 3e^{4\alpha} + 3e^{3\alpha} + e^{2\alpha}$.

- $I_0 = \emptyset$ with probability $\lim_{\alpha\to\infty} e^{3\alpha}/Z = 0$.

- $I_1 = \{Man(Jo)\}$ with probability $\lim_{\alpha\to\infty} e^{3\alpha}/Z = 0$.
- $I_2 = \{Woman(Jo)\}$ with probability $\lim_{\alpha\to\infty} e^{3\alpha}/Z = 0$.
- $I_3 = \{Human(Jo)\}$ is not a stable model of $\overline{KB}_{I_3}$, so its probability is 0.
- $I_4 = \{Man(Jo), Human(Jo)\}$ with probability $\lim_{\alpha\to\infty} e^{4\alpha}/Z = \frac{1}{3}$.
- $I_5 = \{Woman(Jo), Human(Jo)\}$ with probability $\lim_{\alpha\to\infty} e^{4\alpha}/Z = \frac{1}{3}$.
- $I_6 = \{Man(Jo), Woman(Jo)\}$ with probability $\lim_{\alpha\to\infty} e^{2\alpha}/Z = 0$.
- $I_7 = \{Man(Jo), Woman(Jo), Human(Jo)\}$ with probability $\lim_{\alpha\to\infty} e^{4\alpha}/Z = \frac{1}{3}$.

Thus we can check that

- $Pr[Human(Jo) = \mathbf{t}] = Pr[I_4] + Pr[I_5] + Pr[I_7] = 1$: for I_4, KB_3 is disregarded; for I_5, KB_2 is disregarded; for I_7, the last rule of KB_1 is disregarded.
- $Pr[Human(Jo) = \mathbf{t} \mid Man(Jo) = \mathbf{t}] = \frac{Pr[I_4]+Pr[I_7]}{Pr[I_1]+Pr[I_4]+Pr[I_6]+Pr[I_7]} = 1$.
- $Pr[Man(Jo) = \mathbf{t} \mid Human(Jo) = \mathbf{t}] = \frac{Pr[I_4]+Pr[I_7]}{Pr[I_4]+Pr[I_5]+Pr[I_7]} = \frac{2}{3}$.

Often an LP$^{\rm MLN}$ program $\mathbb{P}$ consists of the set $\mathbb{P}^{\rm s}$ of soft rules and the set of $\mathbb{P}^{\rm h}$ of hard rules together, and there exists at least one stable model that is obtained from all hard rules plus some subset of soft rules. In this case, we may simply consider the weights of soft rules only in computing the probabilities of stable models. Let SM$'[\mathbb{P}]$ be the set

$$\{I \mid I \text{ is a stable model of } \overline{\mathbb{P}^{\rm h}} \cup (\overline{\mathbb{P}^{\rm s}})_I\},$$

and let

$$Pr'_{\mathbb{P}}[I] = \begin{cases} \frac{W_{\mathbb{P}^{\rm s}}(I)}{\sum\limits_{J\in {\rm SM}'[\mathbb{P}]} W_{\mathbb{P}^{\rm s}}(J)} & \text{if } I \in {\rm SM}'[\mathbb{P}]; \\ 0 & \text{otherwise.} \end{cases}$$

Note the absence of $\lim\limits_{\alpha\to\infty}$ in the definition of $Pr'_{\mathbb{P}}[I]$. Also unlike $Pr_{\mathbb{P}}[I]$, SM$'[\mathbb{P}]$ may be empty, in which case $Pr'_{\mathbb{P}}[I]$ is not defined.

Proposition 2
Let $\mathbb{P} = \mathbb{P}^{\rm s} \cup \mathbb{P}^{\rm h}$ be an LP$^{\rm MLN}$ program where $\mathbb{P}^{\rm s}$ consists of soft rules and $\mathbb{P}^{\rm h}$ consists of hard rules. If SM$'[\mathbb{P}]$ is not empty, for every interpretation I, $Pr_{\mathbb{P}}[I]$ coincides with $Pr'_{\mathbb{P}}[I]$.

Thus the presence of at least one interpretation in SM$'[\mathbb{P}]$ implies that every other stable model of $\mathbb{P}$ (with non-zero probability) should also satisfy all hard rules in $\mathbb{P}$. Note that Example 1 does not satisfy the nonemptiness condition of SM$'[\mathbb{P}]$, whereas the following example does.

Example 2 (LP$^{\rm MLN}$ *vs. MLN*)
Consider a variant of the main example from (Bauters et al. 2010). We are certain that we booked a concert and that we have a long drive ahead of us unless the concert is cancelled. However, there is a 20% chance that the concert is indeed cancelled. This example can be formalized in LP$^{\rm MLN}$ program $\mathbb{P}$ as

$$\begin{array}{rrcl} \alpha: & ConcertBooked & \leftarrow & \\ \alpha: & LongDrive & \leftarrow & ConcertBooked, not\ Cancelled \\ ln\,0.2: & Cancelled & \leftarrow & \\ ln\,0.8: & & \leftarrow & Cancelled. \end{array}$$

Since SM$'[\mathbb{P}]$ is not empty, in view of Proposition 2, the probability of the two stable models are as follows:

- $I_1 = \{\mathit{ConcertBooked}, \mathit{Cancelled}\}$, with $\mathit{Pr}_{\mathbb{P}}[I_1] = \frac{e^{ln 0.2}}{e^{ln 0.2} + e^{ln 0.8}} = 0.2$.
- $I_2 = \{\mathit{ConcertBooked}, \mathit{LongDrive}\}$, with $\mathit{Pr}_{\mathbb{P}}[I_2] = \frac{e^{ln 0.8}}{e^{ln 0.2} + e^{ln 0.8}} = 0.8$.

If this program is understood under the MLN semantics, say in the syntax

$$\begin{aligned} \alpha &: \mathit{ConcertBooked} \\ \alpha &: \mathit{ConcertBooked} \land \neg \mathit{Cancelled} \rightarrow \mathit{LongDrive} \\ ln\, 0.2 &: \mathit{Cancelled} \\ ln\, 0.8 &: \neg \mathit{Cancelled}, \end{aligned}$$

there are three MLN models with non-zero probabilities:

- $I_1 = \{\mathit{ConcertBooked}, \mathit{Cancelled}\}$ with $\mathit{Pr}[I_1] = 0.2/1.4 \simeq 0.1429$.
- $I_2 = \{\mathit{ConcertBooked}, \mathit{LongDrive}\}$ with $\mathit{Pr}[I_2] = 0.8/1.4 \simeq 0.5714$.
- $I_3 = \{\mathit{ConcertBooked}, \mathit{Cancelled}, \mathit{LongDrive}\}$ with $\mathit{Pr}[I_3] = 0.2/1.4 \simeq 0.1429$.

The presence of I_3 is not intuitive (why have a long drive when the concert is cancelled?)

Remark. In some sense, the distinction between soft rules and hard rules in LP$^{\text{MLN}}$ is similar to the distinction between consistency-restoring rules (CR-rules) and standard ASP rules under CR-Prolog (Balduccini and Gelfond 2003): CR-rules are added to the standard ASP program part until the resulting program has a stable model. CR-Prolog also allows a preference on selecting which CR-rules to be added in order to obtain consistency. In LP$^{\text{MLN}}$ a similar effect can be obtained by adding soft rules with different weights. On the other hand, CR-Prolog has little to say when there is no stable model no matter what CR-rules are added (**c.f.** Example 1).

Remark. This example also illustrates a correspondence between LP$^{\text{MLN}}$ and probabilistic logic programming languages based on the distribution semantics (Sato 1995). The use of soft rules in the example simulates the probabilistic choices under the distribution semantics. However, this correspondence is only valid when there is only one stable model per the probabilistic choice induced by the selection of such soft rules.

Example 3
It is well known that Markov Logic does not properly handle inductive definitions,[2] while LP$^{\text{MLN}}$ gives an intuitive representation. For instance, consider that x may influence y if x is a friend to y, and the influence relation is a minimal relation that is closed under transitivity.

$$\begin{aligned} \alpha &: \mathit{Friend}(A, B) \\ \alpha &: \mathit{Friend}(B, C) \\ 1 &: \mathit{Influences}(x, y) \leftarrow \mathit{Friend}(x, y) \\ \alpha &: \mathit{Influences}(x, y) \leftarrow \mathit{Influences}(x, z), \mathit{Influences}(z, y). \end{aligned}$$

Note that the third rule is soft: a person does not always influence his/her friend. The fourth rule says if x influences z, and z influences y, we can say x influences y. On the other hand, we do not want this relation to be vacuously true.

Assuming that there are only three people A, B, C in the domain (thus there are $1 + 1 + 9 + 27$ ground rules), there are four stable models with non-zero probabilities. Let $Z = e^9 + 2e^8 + e^7$.

[2] "Markov Logic has the drawback that it cannot express (non-ground) inductive definitions." (Fierens et al. 2013)

- $I_1 = \{\mathit{Friend}(A,B), \mathit{Friend}(B,C), \mathit{Influence}(A,B), \mathit{Influence}(B,C), \mathit{Influence}(A,C)\}$ with probability e^9/Z.
- $I_2 = \{\mathit{Friend}(A,B), \mathit{Friend}(B,C), \mathit{Influence}(A,B)\}$ with probability e^8/Z.
- $I_3 = \{\mathit{Friend}(A,B), \mathit{Friend}(B,C), \mathit{Influence}(B,C)\}$ with probability e^8/Z.
- $I_4 = \{\mathit{Friend}(A,B), \mathit{Friend}(B,C)\}$ with probability e^7/Z.

Thus we get

- $\mathit{Pr}_{\mathbb{P}}[\mathit{Influence}(A,B) = \mathbf{t}] = \mathit{Pr}_{\mathbb{P}}[\mathit{Influence}(B,C) = \mathbf{t}] = (e^9 + e^8)/Z = 0.7311.$
- $\mathit{Pr}_{\mathbb{P}}[\mathit{Influence}(A,C) = \mathbf{t}] = e^9/Z = 0.5344.$

Increasing the weight of the soft rule yields higher probabilities for *Influence*$(A,B) = \mathbf{t}$, *Influence*$(B,C) = \mathbf{t}$, *Influence*$(A,C) = \mathbf{t}$. Still, the first two have the same probability, and the third has less probability than the first two.

Note that the minimality of the influence relation is not expressible under the MLN semantics.

3.3 Relating LP$^{\rm MLN}$ *to ASP*

Any logic program under the stable model semantics can be turned into an LP$^{\rm MLN}$ program by assigning the infinite weight to every rule. That is, for any logic program $\Pi = \{R_1, \dots, R_n\}$, the corresponding LP$^{\rm MLN}$ program $\mathbb{P}_\Pi$ is $\{\alpha : R_1, \dots, \alpha : R_n\}$.

Theorem 1
For any logic program Π, the (deterministic) stable models of Π are exactly the (probabilistic) stable models of $\mathbb{P}_\Pi$ whose weight is $e^{k\alpha}$, where k is the number of all (ground) rules in Π. If Π has at least one stable model, then all stable models of $\mathbb{P}_\Pi$ have the same probability, and are thus the stable models of Π as well.

The idea of softening rules in LP$^{\rm MLN}$ is similar to the idea of "weak constraints" in ASP, which is used for certain optimization problems. A weak constraint has the form " $:\sim$ *Body* $[$*Weight* $:$ *Level*$]$." The answer sets of a program Π plus a set of weak constraints are the answer sets of Π which minimize the penalty calculated from *Weight* and *Level* of violated weak constraints. However, weak constraints are more restrictive than weighted rules in LP$^{\rm MLN}$, and do not have a probabilistic semantics.

3.4 Completion: Turning LP$^{\rm MLN}$ *to MLN*

It is known that the stable models of a tight logic program coincide with the models of the program's completion. This yielded a way to compute stable models using SAT solvers. The method can be extended to LP$^{\rm MLN}$ so that their stable models along with the probability distribution can be computed using existing implementations of MLNs, such as Alchemy (`http://alchemy.cs.washington.edu`) and Tuffy (`http://i.stanford.edu/hazy/hazy/tuffy`).

We define the *completion* of $\mathbb{P}$, denoted *Comp*$(\mathbb{P})$, to be the MLN which is the union of $\mathbb{P}$ and the hard rules

$$\alpha : \ A \rightarrow \bigvee_{\substack{w:A_1,\dots,A_k \leftarrow Body \ \in \ \mathbb{P} \\ A \in \{A_1,\dots,A_k\}}} \left(Body \wedge \bigwedge_{A' \in \{A_1,\dots,A_k\} \setminus \{A\}} \neg A' \right)$$

for each ground atom A.

This is a straightforward extension of the completion from (Lee and Lifschitz 2003) simply assigning the infinite weight α to the completion formulas. Likewise, we say that LP$^{\text{MLN}}$ program $\mathbb{P}$ is *tight* if $\overline{\mathbb{P}}$ is tight according to (Lee and Lifschitz 2003).

Theorem 2
For any tight LP$^{\text{MLN}}$ program $\mathbb{P}$ such that SM$'[\mathbb{P}]$ is not empty, $\mathbb{P}$ (under the LP$^{\text{MLN}}$ semantics) and *Comp*$(\mathbb{P})$ (under the MLN semantics) have the same probability distribution over all interpretations.

4 Embedding Pearl's Probabilistic Causal Models in LP$^{\text{MLN}}$

4.1 Review: Pearl's Causal Models

Notation: Following (Pearl 2000), we use capital letters (e.g., X, Y, Z, U, V) for (lists of) atoms and lower case letters (x, y, z, u, v) for generic symbols for specific (lists of) truth values taken by the corresponding (lists of) atoms. When X is a list, we use subscripts, such as X_i, to denote an element in X.

As usual, a propositional formula is constructed from atoms, **t**, **f**, and propositional connectives, $\neg$, $\wedge$, $\vee$, $\rightarrow$.

Definition 1 (*structural theory*)
Assume that a finite set of propositional atoms is partitioned into a set of exogenous atoms U and a set of endogenous atoms $V = \{V_1, \ldots, V_n\}$. A *Boolean structural theory* is $\langle U, V, F \rangle$, where F is a finite set of equations $V_i = F_i$, one for each endogenous atom V_i, and F_i is a propositional formula.

Definition 2 (*causal diagram*)
The *causal diagram* of a Boolean structural theory $\langle U, V, F \rangle$ is the directed graph whose vertices are the atoms in $U \cup V$ and an edge goes from V_j to V_i if there is an equation $V_i = F_i$ in the structural theory such that V_j occurs in F_i. We say that the structural theory is *acyclic* if its causal diagram is acyclic.

For any interpretation I and J of $U \cup V$, we say that $J \neq^V I$ if J and I agree on all atoms in U and do not agree on some atoms in V.

Definition 3 (*solution*)
Given a Boolean causal theory $\langle U, V, F \rangle$, a *solution* (or a *causal world*) I is any interpretation of $U \cup V$ such that

- I satisfies the equivalences $V_i \leftrightarrow F_i$ for all equations $V_i = F_i$ in F, and
- no other interpretation J such that $J \neq^V I$ satisfies all such equivalences $V_i \leftrightarrow F_i$.

Definition 4 (*causal model*)
A *(Boolean) causal model* $\langle U, V, F \rangle$ is an acyclic Boolean structural theory that has a unique solution for each realization (i.e., truth assignment) of U; in other words, each truth assignment of U has a unique expansion to $U \cup V$ that is a solution.

Definition 5 (*probabilistic causal model*)
A *probabilistic (Boolean) structural theory* is a pair

$$\langle \langle U, V, F\rangle, P(U)\rangle \tag{3}$$

where $\langle U, V, F\rangle$ is a Boolean structural theory, and $P(U)$ is a probability distribution over U. We assume that exogenous atoms are independent of each other. A *Probabilisitic (Boolean) Causal Model (PCM)* is a probabilistic structural theory (3) such that $\langle U, V, F\rangle$ is a causal model. The *solution*s of PCM (3) are the solutions of $\langle U, V, F\rangle$. The probability of a solution I under the PCM $\mathbb{M}$, denoted $P_{\mathbb{M}}(I)$, is defined as $P(U = I(U))$.

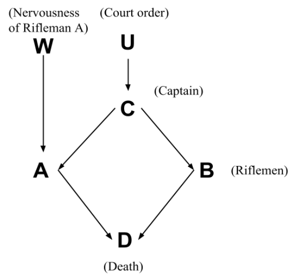
Given a PCM $\mathbb{M} = \langle\langle U, V, F\rangle, P(U)\rangle$, for any subset Y of V, we write $Y_{\mathbb{M}}(u)$ to denote the truth assignment of Y in the solution of $\mathbb{M}$ induced by u. The probability of $Y = y$ is defined as

$$P_{\mathbb{M}}(Y = y) = \textstyle\sum_{\{u \mid Y_{\mathbb{M}}(u)=y\}} P(u).$$

For any subset Y, Z of V, $P_{\mathbb{M}}(Y = y \mid Z = z)$ is defined as

$$P_{\mathbb{M}}(Y = y \mid Z = z) = \frac{\sum_{\{u \ \mid \ Y_{\mathbb{M}}(u) \, = \, y \text{ and } Z_{\mathbb{M}}(u) \, = \, z\}} P(u)}{\sum_{\{u \mid Z_{\mathbb{M}}(u)=z\}} P(u)}.$$

Consider, for example, the probabilistic causal model $\mathbb{M}_{FS}$ for the Firing Squad example (Pearl 2000, Sec 7.1.2):



$$\begin{array}{rl} \mathbb{M}_{FS} = & \langle\langle\{U, W\}, \{C, A, B, D\}, F\rangle, P(U, W)\rangle \\ F: & C = U \\ & A = C \vee W \\ & B = C \\ & D = A \vee B \\ & \\ & P(U{=}\mathbf{t}) = p \\ & P(W{=}\mathbf{t}) = q \end{array}$$

U denotes "The court orders the execution," C denotes "The captain gives a signal," A denotes "Rifleman A shoots," B denotes "Rifleman B shoots," D denotes "The prisoner dies," and W denotes "Rifleman A is nervous." There is a probability p that the court has ordered the execution; rifleman A has a probability q of pulling the trigger out of nervousness. The PCM has four solutions for each realization of U and W.

Solutions	Probability
$\{U = \mathbf{f}, W = \mathbf{f}, C = \mathbf{f}, A = \mathbf{f}, B = \mathbf{f}, D = \mathbf{f}\}$	$(1-p)(1-q)$
$\{U = \mathbf{f}, W = \mathbf{t}, C = \mathbf{f}, A = \mathbf{t}, B = \mathbf{f}, D = \mathbf{t}\}$	$(1-p)q$
$\{U = \mathbf{t}, W = \mathbf{f}, C = \mathbf{t}, A = \mathbf{t}, B = \mathbf{t}, D = \mathbf{t}\}$	$p(1-q)$
$\{U = \mathbf{t}, W = \mathbf{t}, C = \mathbf{t}, A = \mathbf{t}, B = \mathbf{t}, D = \mathbf{t}\}$	pq

4.2 Embedding PCM in LP$^{\text{MLN}}$

Since causal models assume propositional formulas, it is convenient to discuss the result by first extending the syntax of LP$^{\text{MLN}}$ to weighted propositional formulas, that is of the form

$w : F$ where F is a propositional formula and w is either a real number or the symbol α. We refer the reader to (Ferraris 2005) for the definition of a stable model for propositional formulas. Extending LP$^{\rm MLN}$ to this general syntax is straightforward, which we skip due to lack of space.

Definition 6 (LP$^{\rm MLN}$ *representation*)
For any PCM $\mathbb{M} = \langle\langle U, V, F\rangle, P(U)\rangle$, $\mathbb{P}_{\mathbb{M}}$ is the LP$^{\rm MLN}$ program consisting of

- $\alpha : \; V_i \leftarrow F_i$ for each equation $V_i = F_i$ in $\mathbb{M}$, and,
- for each exogenous atom U_i of $\mathbb{M}$ such that $P(U_i = \mathbf{t}) = p$: (i) $ln(p) : \; U_i$ and $ln(1-p) : \leftarrow U_i$ if $0 < p < 1$; (ii) $\alpha : \; U_i$ if $p = 1$; (iii) $\alpha : \leftarrow U_i$ if $p = 0$.

For the Firing Squad example, assuming $0 < p, q < 1$, $\mathbb{P}_{\mathbb{M}_{FS}}$ is as follows:

$$\begin{array}{rlcrl} ln(p) : & U & \qquad & \alpha : & C \leftarrow U \\ ln(1-p) : & \leftarrow U & & \alpha : & A \leftarrow C \vee W \\ ln(q) : & W & & \alpha : & B \leftarrow C \\ ln(1-q) : & \leftarrow W & & \alpha : & D \leftarrow A \vee B. \end{array}$$

Theorem 3
The solutions of a probabilistic causal model $\mathbb{M}$ are identical to the stable models of $\mathbb{P}_{\mathbb{M}}$ and their probability distributions coincide.

Note that the acyclicity condition in PCM implies the tightness condition of its LP$^{\rm MLN}$ program representation. Thus, using the completion method in Section 3.4, we can automate query answering for this domain using Alchemy.

4.3 Review: Counterfactuals in PCM

Definition 7 (*submodel*)
Given a Boolean causal model $M = \langle U, V, F\rangle$, and a subset X of V, the *submodel* $M_{X=x}$ of M is the Boolean causal model obtained from M by replacing every equation $X_i = F_i$ in the theory, where $X_i \in X$, with $X_i = x_i$. Given a PCM $\mathbb{M} = \langle M, P(U)\rangle$, $\mathbb{M}_{X=x} = \langle M_{X=x}, P(U)\rangle$.

For any PCM $\mathbb{M} = \langle\langle U, V, F\rangle, P(U)\rangle$, let X, Y, Z be subsets of V. The probability of a counterfactual statement, represented as $Y_{X=x} = y$, is defined as

$$P_{\mathbb{M}}(Y_{X=x} = y) = \sum_{\{u \mid Y_{\mathbb{M}_{X=x}}(u)=y\}} P(u).$$

The probability of a conditional counterfactual statement "Given Z is z, Y would have been y had X been x", represented as $Y_{X=x} = y \mid Z = z$, is defined as

$$P_{\mathbb{M}}(Y_{X=x} = y \mid Z = z) = \frac{\sum_{\{u \;\mid\; Y_{\mathbb{M}_{X=x}}(u) \;=\; y \text{ and } Z_{\mathbb{M}}(u) \;=\; z\}} P(u)}{\sum_{\{u \mid Z_{\mathbb{M}}(u)=z\}} P(u)}.$$

For example, given that the prisoner is dead, what is the probability that the prisoner were not dead if rifleman A had not shot? This is asking: $P_{\mathbb{M}}(D_{A=\mathbf{f}} = \mathbf{f} \mid D = \mathbf{t}) = \frac{(1-p)q}{1-(1-p)(1-q)}$.

4.4 PCM Counterfactuals in LP$^{\rm MLN}$

Counterfactual reasoning in PCM can be turned into LP$^{\rm MLN}$ reasoning. The program LP$^{\rm MLN}$ $\mathbb{P}_{\mathbb{M}}^{\rm twin}$ consists of

- all rules in $\mathbb{P}_{\mathbb{M}}$;
- rule
$$\alpha: \quad V_i^* \ \leftarrow\ F_i^* \wedge \neg Do(V_i{=}\mathbf{t}) \wedge \neg Do(V_i{=}\mathbf{f})$$
for each equation $V_i = F_i$ in $\mathbb{M}$, where V_i^* is a new symbol corresponding to V_i, and F_i^* is a formula obtained from F by replacing every occurrence of endogenous atoms W with W^*.
- rule
$$\alpha: \quad V_i^* \leftarrow Do(V_i{=}\mathbf{t}) \tag{4}$$
for every $V_i \in V$.

(Note that $Do(V_i{=}\mathbf{t})$ is an atom, containing "=" as a part of the string.)

Theorem 4
For any PCM $\mathbb{M} = \langle\langle U, V, F\rangle, P(U)\rangle$ and any subsets X, Y, Z of V, which are not necessarily disjoint from each other,

$$P_{\mathbb{M}}(Y_{X=x} = y \mid Z = z) \ = \ Pr_{\mathbb{P}_{\mathbb{M}}^{\rm twin} \cup Do(X=x)}[Y^* = y \mid Z = z],$$

where $Do(X = x)$ is $\{\alpha : Do(X_i = x_i) \mid i = 1, \ldots, |X|\}$.

Readers who are familiar with the twin network method for counterfactual reasoning (Balke and Pearl 1994) would notice that $\mathbb{P}_{\mathbb{M}}^{\rm twin} \cup Do(X = x)$ represents the twin network obtained from $\mathbb{M}$, where starred atoms represent the counterfactual world.

For the Firing Squad example, $\mathbb{P}_{\mathbb{M}_{FS}}^{\rm twin}$ is the union of $\mathbb{P}_{\mathbb{M}_{FS}}$ and the set of rules

$$\begin{array}{ll}
\alpha: & C^* \leftarrow U, not\ Do(C = \mathbf{t}), not\ Do(C = \mathbf{f})\\
\alpha: & A^* \leftarrow C^*, not\ Do(A = \mathbf{t}), not\ Do(A = \mathbf{f})\\
\alpha: & A^* \leftarrow W, not\ Do(A = \mathbf{t}), not\ Do(A = \mathbf{f})\\
\alpha: & B^* \leftarrow C^* \wedge \neg Do(B = \mathbf{t}) \wedge \neg Do(B = \mathbf{f})\\
\alpha: & D^* \leftarrow (A^* \vee B^*) \wedge \neg Do(D = \mathbf{t}) \wedge \neg Do(D = \mathbf{f})
\end{array}$$

and rules (4) for $V_i \in \{C, A, B, D\}$. In accordance with Theorem 4,

$$Pr_{\mathbb{P}_{\mathbb{M}_{FS}}^{\rm twin} \cup \{\alpha : Do(A=\mathbf{f})\}}[D^*{=}\mathbf{f} \mid D{=}\mathbf{t}] = \frac{(1-p)q}{1-(1-p)(1-q)}.$$

The LP$^{\rm MLN}$ representation is similar to the one in (Baral and Hunsaker 2007), which turns PCM into P-log. However, the LP$^{\rm MLN}$ representation is simpler; we require neither auxiliary predicates, such as $intervene$ and obs, nor strong negation.

4.5 Other Related Work

In (Lee and Wang 2015) it is shown that a version of ProbLog from (Fierens et al. 2013) can be embedded in LP$^{\rm MLN}$. This result can be extended to embed Logic Programs with Annotated Disjunctions (LPAD) in LP$^{\rm MLN}$ based on the fact that any LPAD program can be further turned into a ProbLog program by eliminating disjunctions in the heads (Gutmann 2011, Section 3.3).

It is known that LPAD is related to several other languages. In (Vennekens et al. 2004), it is shown that Poole's ICL (Poole 1997) can be viewed as LPAD, and that acyclic LPAD programs can be turned into ICL. This indirectly tells us how ICL can be embedded in $\text{LP}^{\rm MLN}$.

CP-logic (Vennekens et al. 2009) is a probabilistic extension of FO(ID) (Denecker and Ternovska 2007). It is shown in (Vennekens et al. 2006), that CP-logic "almost completely coincides" with LPAD.

P-log (Baral et al. 2009) is another language whose logical foundation is answer set programs. Like $\text{LP}^{\rm MLN}$, it considers the possible worlds to be answer sets, which represent an agent's rational beliefs, rather than any interpretations. The difference is that P-log's probabilistic foundation is Causal Bayesian Networks, whereas Markov Logic serves as the probabilistic foundation of $\text{LP}^{\rm MLN}$. P-log is distinct from other earlier work in that it allows for expressing probabilistic nonmonotonicity, the ability of the reasoner to change its probabilistic model as a result of new information. However, inference in the implementation of P-log is not scalable as it has to enumerate all stable models.

PrASP (Nickles and Mileo 2014) is a recent language similar to $\text{LP}^{\rm MLN}$ in that the probability distribution is obtained from the weights of the formulas. In addition, (Ng and Subrahmanian 1994) and (Saad and Pontelli 2005) introduce other probabilistic extensions of stable model semantics.

While the study of more precise relationships between $\text{LP}^{\rm MLN}$ and the above languages is future work, one notable distinction is that $\text{LP}^{\rm MLN}$ uses Markov Logic as a monotonic basis, similar to the way ASP uses SAT as a monotonic basis. Most of the languages are meaningful only when the knowledge base is consistent, and thus do not address the inconsistency handling as in $\text{LP}^{\rm MLN}$.

5 Conclusion

$\text{LP}^{\rm MLN}$ is a simple, intuitive approach to combine both ASP and MLNs. $\text{LP}^{\rm MLN}$ provides a simple solution to inconsistency handling in ASP, especially when ASP knowledge bases are combined from different sources.

While MLN is an undirected approach, $\text{LP}^{\rm MLN}$ is a directed approach, where the directionality comes from the stable model semantics. This makes $\text{LP}^{\rm MLN}$ closer to Pearl's causal models and ProbLog.

The work presented here calls for more future work. Obviously, there are many existing languages that we did not formally compare with $\text{LP}^{\rm MLN}$. While a fragment of $\text{LP}^{\rm MLN}$ can be computed by existing implementations of and MLNs, one may design a native computation method for the general case. The close relationship between $\text{LP}^{\rm MLN}$ and MLNs may tell us how to apply machine learning methods developed for MLNs to work with $\text{LP}^{\rm MLN}$ programs.

Acknowledgements We are grateful to Michael Bartholomew, Amelia Harrison, and the anonymous referees for their useful comments. This work was partially supported by the National Science Foundation under Grant IIS-1319794, South Korea IT R&D program MKE/KIAT 2010-TD-300404-001, and ICT R&D program of MSIP/IITP 10044494 (WiseKB).

References

BALDUCCINI, M. AND GELFOND, M. 2003. Logic programs with consistency-restoring rules[3]. In *Working Notes of the AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*.

BALKE, A. AND PEARL, J. 1994. Counterfactual probabilities: Computational methods, bounds and applications. In *Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 46–54.

BARAL, C., GELFOND, M., AND RUSHTON, J. N. 2009. Probabilistic reasoning with answer sets. *TPLP 9,* 1, 57–144.

BARAL, C. AND HUNSAKER, M. 2007. Using the probabilistic logic programming language P-log for causal and counterfactual reasoning and non-naive conditioning. In *IJCAI*. 243–249.

BAUTERS, K., SCHOCKAERT, S., DE COCK, M., AND VERMEIR, D. 2010. Possibilistic answer set programming revisited. In *26th Conference on Uncertainty in Artificial Intelligence (UAI 2010)*.

BOCHMAN, A. AND LIFSCHITZ, V. 2015. Pearl's causality in a logical setting. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

DENECKER, M. AND TERNOVSKA, E. 2007. Inductive situation calculus. *Artificial Intelligence 171,* 5-6, 332–360.

FABER, W., LEONE, N., AND PFEIFER, G. 2004. Recursive aggregates in disjunctive logic programs: Semantics and complexity. In *Proceedings of European Conference on Logics in Artificial Intelligence (JELIA)*.

FERRARIS, P. 2005. Answer sets for propositional theories. In *Proceedings of International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)*. 119–131.

FERRARIS, P., LEE, J., LIERLER, Y., LIFSCHITZ, V., AND YANG, F. 2012. Representing first-order causal theories by logic programs. *TPLP 12,* 3, 383–412.

FIERENS, D., VAN DEN BROECK, G., RENKENS, J., SHTERIONOV, D., GUTMANN, B., THON, I., JANSSENS, G., AND DE RAEDT, L. 2013. Inference and learning in probabilistic logic programs using weighted boolean formulas. *Theory and Practice of Logic Programming*, 1–44.

GUTMANN, B. 2011. On continuous distributions and parameter estimation in probabilistic logic programs. Ph.D. thesis, KU Leuven.

LEE, J. 2005. A model-theoretic counterpart of loop formulas. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*. Professional Book Center, 503–508.

LEE, J. AND LIFSCHITZ, V. 2003. Loop formulas for disjunctive logic programs. In *Proceedings of the 19th International Conference on Logic Programming (ICLP)*. 451–465.

LEE, J. AND WANG, Y. 2015. A probabilistic extension of the stable model semantics. In *International Symposium on Logical Formalization of Commonsense Reasoning, AAAI 2015 Spring Symposium Series*. To appear.

LIN, F. AND ZHAO, Y. 2004. ASSAT: Computing answer sets of a logic program by SAT solvers. *Artificial Intelligence 157*, 115–137.

MCCAIN, N. AND TURNER, H. 1997. Causal theories of action and change. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*. 460–465.

NG, R. AND SUBRAHMANIAN, V. 1994. Stable semantics for probabilistic deductive databases. *Information and computation 110,* 1, 42–83.

NICKLES, M. AND MILEO, A. 2014. Probabilistic inductive logic programming based on answer set programming.

NIEMELÄ, I. AND SIMONS, P. 2000. Extending the Smodels system with cardinality and weight constraints. In *Logic-Based Artificial Intelligence*, J. Minker, Ed. Kluwer, 491–521.

PEARL, J. 2000. *Causality: models, reasoning and inference*. Vol. 29. Cambridge Univ Press.

[3] `http://www.krlab.cs.ttu.edu/papers/download/bg03.pdf`

PELOV, N., DENECKER, M., AND BRUYNOOGHE, M. 2007. Well-founded and stable semantics of logic programs with aggregates. *TPLP 7,* 3, 301–353.

POOLE, D. 1997. The independent choice logic for modelling multiple agents under uncertainty. *Artificial Intelligence 94*, 7–56.

SAAD, E. AND PONTELLI, E. 2005. Hybrid probabilistic logic programming with non-monotoic negation. In *Proceedings of the 21st International Conference on Logic Programming (ICLP).*

SATO, T. 1995. A statistical learning method for logic programs with distribution semantics. In *Proceedings of the 12th International Conference on Logic Programming (ICLP)*. 715–729.

SON, T. C., PONTELLI, E., AND TU, P. H. 2006. Answer sets for logic programs with arbitrary abstract constraint atoms. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence (AAAI).*

VENNEKENS, J., DENECKER, M., AND BRUYNOOGHE, M. 2006. Representing causal information about a probabilistic process. In *Logics In Artificial Intelligence*. 452–464.

VENNEKENS, J., DENECKER, M., AND BRUYNOOGHE, M. 2009. CP-logic: A language of causal probabilistic events and its relation to logic programming. *TPLP 9,* 3, 245–308.

VENNEKENS, J., VERBAETEN, S., BRUYNOOGHE, M., AND A, C. 2004. Logic programs with annotated disjunctions. In *Proceedings of the 20th International Conference on Logic Programming (ICLP)*. 431–445.