

Zugzwang

Stochastic Adventures in Inductive Logic

Francisco Coelho

Departamento de Informática, Universidade de Évora
High Performance Computing Chair
NOVA-LINCS

November 29, 2022

- 1 Introduction
- 2 Extending Probability to Samples
- 3 Cases & Examples
- 4 Conclusions

Notation and Assumptions

- $\bar{x} = 1 - x$.
- **Probabilistic Atomic Choice (PAC):** $x :: a$ defines $a \vee \neg a$ and probabilities $p(a) = x, p(\neg a) = \bar{x}$.
- δa denotes $a \vee \neg a$ and $\delta\{x :: a, a \in \mathcal{A}\} = \{\delta a, a \in \mathcal{A}\}$ for a set of atoms \mathcal{A} .
- **Closed World Assumption:** $\sim p \models \neg p$.

General Setting

- **Atoms** \mathcal{A} , $\overline{\mathcal{A}} = \{\neg a, a \in \mathcal{A}\}$, and **literals** $\mathcal{L} = \mathcal{A} \cup \overline{\mathcal{A}}$.
- **Samples** $z \in \mathcal{Z} \iff z \subseteq \mathcal{L}$.
- **Events** or *consistent samples* \mathcal{E} :

$$\mathcal{E} = \{z \in \mathcal{Z}, \forall a \in \mathcal{A} \mid \{a, \neg a\} \cap z \mid \leq 1\}.$$

- **PASP Problem** or **Specification**: $P = C \wedge F \wedge R$ where
 - $C = C_P = \{x_i :: a_i, i \in 1 : n \wedge a_i \in \mathcal{A}\}$ *pacs*.
 - $F = F_P$ *facts*.
 - $R = R_P$ *rules*.
 - $\mathcal{A}_P, \mathcal{Z}_P$ and \mathcal{E}_P : *atoms, samples and events of P*.
- **Stable Models** of P , $S = S_P$, are the stable models of $\delta P = \delta C + F + R$.

Distribution Semantics

- **Total Choices:** $\Theta = \Theta_C = \Theta_P$ elements are $\theta = \{t_c, c \in C\}$ where $c = x :: a$ and t_c is a or $\neg a$.
- **Total Choice Probability:**

$$p(\theta) = \prod_{a \in \theta} x \prod_{\neg a \in \theta} \bar{x}. \quad (1)$$

This is the *distribution semantic* as set by Sato.

Problem Statement

How to *extend* probability from total choices to stable models, events and samples?

There's a problem right at extending to stable models.

The Disjunction Case

Disjunction Example

The specification

$$0.3 :: a,$$
$$b \vee c \leftarrow a.$$

has three stable models,

$$s_1 = \{\neg a\}, \quad s_2 = \{a, b\}, \quad s_3 = \{a, c\}.$$

- Any stable model contains exactly one total choice. ■
- $p(\{\neg a\}) = 0.7$ is straightforward.
- But, no *informed* choice for $x \in [0, 1]$ in

$$p(\{a, b\}) = 0.3x,$$

$$p(\{a, c\}) = 0.3\bar{x}.$$

Lack of Information & Parametrization

- The specification *lacks information* to set $x \in [0, 1]$ in

$$p(\{a, b\}) = 0.3x,$$

$$p(\{a, c\}) = 0.3\bar{x}.$$

- A *random variable* captures this uncertainty, **assuming** that the stable models are statistically independent:

$$p(\{\neg a\} \mid X = x) = 0.7,$$

$$p(\{a, b\} \mid X = x) = 0.3x,$$

$$p(\{a, c\} \mid X = x) = 0.3\bar{x}.$$

- Other uncertainties may lead to further conditions:

$$p(s \mid X_1 = x_1, \dots, X_n = x_n).$$

Reducing **uncertainty**, e.g. setting $X = 0.21$, must result from **external** sources, since the specification lacks information for further assertions.

Independence of Stable Models

Q: Why are the stable models assumed statistically independent?

A: Because dependence can be *explicitly* modelled.

- So, it is assumed *intention* of the *modeller* to not explicitly express such dependences.
- **For example:** **TODO** Some key examples.

A *random variable* captures this uncertainty:

$$\begin{aligned}p(\{\neg a\} \mid X = x) &= 0.7, \\p(\{a, b\} \mid X = x) &= 0.3x, \\p(\{a, c\} \mid X = x) &= 0.3\bar{x}.\end{aligned}$$

Main Research Question

Can *all* specification uncertainties be neatly expressed as that example?

- Follow ASP syntax; for each case, what are the uncertainty scenarios?
- The disjunction example illustrates one such scenario.
- *Neat* means a function $d : \mathcal{S} \rightarrow [0, 1]$ such that

$$\sum_{s \in \mathcal{S}_\theta} d(s) = 1$$

for each $\theta \in \Theta$.

Leap into Inductive Programming

Given a method that produces a distribution of samples, p , from a specification, P and:

- Z , a dataset (of samples).
- e , the respective empirical distribution.
- D , some probability divergence, e.g. Kullback-Leibler.

Specification Performance & Inductive Programming

- $D(P) = D(e, p)$ is a **performance** measure of P .
- Predictor performance measures, such as accuracy, are common in *Machine Learning* tasks.
- For *Inductive Programming* this performance can be used, e.g. as fitness, by algorithms searching for **optimal specifications of a dataset**.

- 1 Introduction
- 2 Extending Probability to Samples**
- 3 Cases & Examples
- 4 Conclusions

Resolution Path

Prior to *conciliation* with data:

- ① **Hopefully**, *conditional parameters* extend probability from total choices to *standard models*.
- ② **How** to extend it to *events*?
 - $p(x) = 0$ for x *excluded* by the specification, including *inconsistent* samples.
 - $p(x)$ depends on the $s \in \mathcal{S}$ that contain/are contained in x .

Consider probabilities **conditional** on the total choice!

Bounds of Events

- For $x \in \mathcal{E}$:
 - **Lower Models:** $\langle x | = \{s \in \mathcal{S}, s \subseteq x\}$.
 - **Upper Models:** $|x \rangle = \{s \in \mathcal{S}, x \subseteq s\}$.
- **Proposition.** Exactly *one* of the following cases takes place:

- ① $\langle x | = \{x\} = |x \rangle$ and x is a stable model. Then:

$$p(x \mid C = \theta_x) = d(x). \quad (2)$$

- ② $\langle x | \neq \emptyset \wedge |x \rangle = \emptyset$. Then:

$$p(x \mid C = \theta_s, s \in \langle x |) = \prod_{s \in \langle x |} d(s). \quad (3)$$

- ③ $\langle x | = \emptyset \wedge |x \rangle \neq \emptyset$. Then:

$$p(x \mid C = \theta_s, s \in |x \rangle) = \sum_{s \in |x \rangle} d(s). \quad (4)$$

- ④ $\langle x | = \emptyset = |x \rangle$. Then:

$$p(x) = 0. \quad (5)$$

because stable models are *minimal*.

Conditional on Total Choices

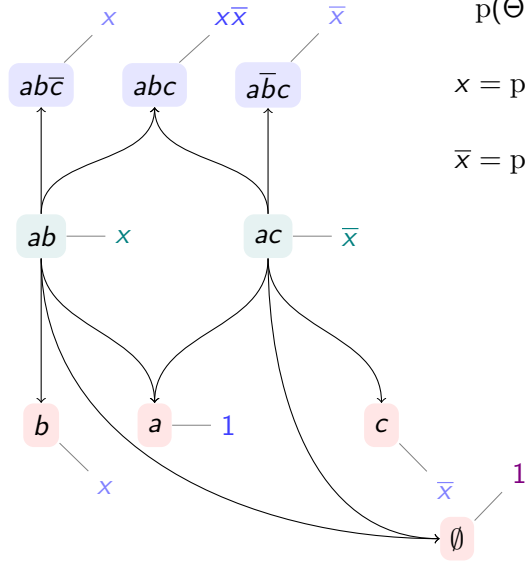
- A stable model is entailed by an atomic choice plus the facts and rules of the specification.
- We express that entailment as a *conditional*. For example:

$$p(\{a, b\} \mid X = x) = p(b \mid X = x, \Theta = a) p(\theta = a)$$

- And now $p(b \mid X = x, \Theta = a) = x$, since X is a proxy for the stable models of the total choice $\theta = a$, we can further.

Disjunction Example | The Events Lattice

$$p(E = abc \mid \Theta) = p(S = ab, S = ac \mid \Theta)$$
$$p(\Theta = a) = 0.3$$

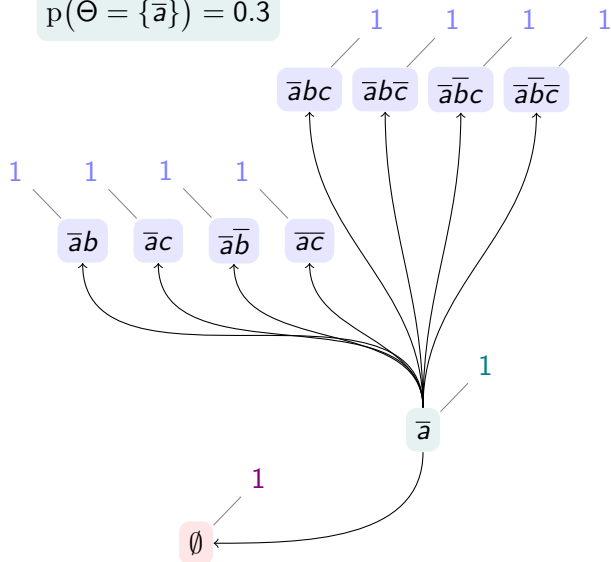


$$x = p(S = ab \mid \Theta)$$

$$\bar{x} = p(S \neq ab \mid \Theta)$$

Disjunction Example | The Events Lattice

$$p(\Theta = \{\bar{a}\}) = \overline{0.3}$$



- Consider the ASP program $P = C \wedge F \wedge R$ with total choices Θ and stable models \mathcal{S} .
- Let $d : \mathcal{S} \rightarrow [0, 1]$ such that $\sum_{s \in \mathcal{S}_\theta} d(s) = 1$ for each $\theta \in \Theta$.

For each $z \in \mathcal{Z}$ only one of the following cases takes place

- ① z is inconsistent. Then **define**

$$w_d(x) = 0. \quad (6)$$

- ② z is an event and $\langle z | = \{z\} = |z\rangle$. Then z is a stable model and **define**

$$w_d(z) = w(z) = d(z) p(\theta_z). \quad (7)$$

- ③ z is an event and $\langle z | \neq \emptyset \wedge |z\rangle = \emptyset$. Then **define**

$$w_d(z) = \sum_{s \in \langle z |} w_d(s). \quad (8)$$

- ④ z is an event and $\langle z | = \emptyset \wedge |z\rangle \neq \emptyset$. Then **define**

$$w_d(z) = \prod_{s \in |z\rangle} w_d(s). \quad (9)$$

- ⑤ z is an event and $\langle z | = \emptyset \wedge |z\rangle = \emptyset$. Then **define**

$$w_d(z) = 0. \quad (10)$$

- ① The last point defines a “weight” function on the samples that depends not only on the total choices and stable models of a PASP but also on a certain function d that must respect some conditions. To simplify the notation we use the subscript in w_d only when necessary.
- ② At first, it may seem counter-intuitive that $w(\emptyset) = \sum_{s \in \mathcal{S}} w(s)$ is the largest “weight” in the lattice. But \emptyset , as an event, sets zero restrictions on the “compatible” stable models. The “complement” of $\perp = \emptyset$ is the *maximal inconsistent* sample $\top = \mathcal{A} \cup \{\neg a, a \in \mathcal{A}\}$.
- ③ **We haven't yet defined a probability measure.** To do so we must define a set of samples Ω , a set of events $F \subseteq \mathbb{P}(\Omega)$ and a function $P : F \rightarrow [0, 1]$ such that:
 - ① $p(E) \in [0, 1]$ for any $E \in F$.
 - ② $p(\Omega) = 1$.
 - ③ if $E_1 \cap E_2 = \emptyset$ then $p(E_1 \cup E_2) = p(E_1) + p(E_2)$.
- ④ In the following, assume that the stable models are iid.
- ⑤ Let the sample space $\Omega = \mathcal{Z}$ and the event space $F = \mathbb{P}(\Omega)$. Define $Z = \sum_{\zeta \in \mathcal{Z}} w(\zeta)$ and

$$p(z) = \frac{w(z)}{Z} \quad z \in \Omega \quad (11)$$

① Introduction

② Extending Probability to Samples

③ **Cases & Examples**

Programs with disjunctive heads

Non-stratified programs

④ Conclusions

- 1 Introduction
- 2 Extending Probability to Samples
- 3 Cases & Examples
 - Programs with disjunctive heads
 - Non-stratified programs
- 4 Conclusions

Consider the program:

$$c_1 = a \vee \neg a,$$

$$c_2 = b \vee c \leftarrow a.$$

This program has two total choices,

$$\theta_1 = \{\neg a\},$$

$$\theta_2 = \{a\}.$$

and three stable models,

$$s_1 = \{\neg a\},$$

$$s_2 = \{a, b\},$$

$$s_3 = \{a, c\}.$$

Suppose that we add an annotation $x :: a$, which entails $\bar{x} :: \neg a$. This is enough to get $w(s_1) = \bar{x}$ but, on the absence of further information, no fixed probability can be assigned to either model s_2, s_3 except that the respective sum must be x . So, expressing our lack of knowledge using a parameter $d \in [0, 1]$ we get:

$$\begin{cases} w(s_1) = \bar{x} \\ w(s_2) = dx \\ w(s_3) = \bar{d}x. \end{cases}$$

In this diagram:

- Negations are represented as e.g. \bar{a} instead of $\neg a$; Stable models are denoted by shaded nodes as ab .
- Events in $\langle x|$ are e.g. a and those in $|x\rangle$ are e.g. \bar{a} . The remaining are simply denoted by e.g. $a\bar{b}$.
- The edges connect stable models with related events. Up arrow indicate links to $|s\rangle$ and down arrows to $\langle s|$.
- The *weight propagation* sets:

$$w(abc) = w(ab)w(ac) = x^2d\bar{d},$$

$$w(\bar{a} \cdot \cdot) = w(\neg a) = \bar{x},$$

$$w(a) = w(ab) + w(ac) = x(d + \bar{d}) = x,$$

$$w(b) = w(ab) = dx,$$

$$w(c) = w(ac) = \bar{d}x,$$

$$w(\emptyset) = w(ab) + w(ac) + w(\neg a) = dx + \bar{d}x + \bar{x} = 1,$$

$$w(a\bar{b}) = 0.$$

- The total weight is

- ① Introduction
- ② Extending Probability to Samples
- ③ Cases & Examples
 - Programs with disjunctive heads
 - Non-stratified programs**
- ④ Conclusions

The following LP is non-stratified, because has a cycle with negated arcs:

$$c_1 = a \vee \neg a,$$

$$c_2 = b \leftarrow \sim c \wedge \sim a,$$

$$c_3 = c \leftarrow \sim b.$$

This program has three stable models

$$s_1 = \{a, c\},$$

$$s_2 = \{\neg a, b\},$$

$$s_3 = \{\neg a, c\}.$$

The disjunctive clause $a \vee \neg a$ defines a set of **total choices**

$$\Theta = \{\theta_1 = \{a\}, \theta_2 = \{\neg a\}\}.$$

Looking into probabilistic events of the program and/or its models, we define $x = p(\Theta = \theta_1) \in [0, 1]$ and $p(\Theta = \theta_2) = \bar{x}$.

Since s_1 is the only stable model that results from $\Theta = \theta_1$, it is natural to extend $p(s_1) = p(\Theta = \theta_1) = x$. However, there is no clear way to assign $p(s_2), p(s_3)$ since *both models result from the single total choice* $\Theta = \theta_2$. Clearly,

$$p(s_2 | \Theta) + p(s_3 | \Theta) = \begin{cases} 0 & \text{if } \Theta = \theta_1 \\ 1 & \text{if } \Theta = \theta_2 \end{cases}$$

but further assumptions are not supported *a priori*. So let's **parameterize** the equation above,

$$\begin{cases} p(s_2 | \Theta = \theta_2) = \beta \in [0, 1] \\ p(s_3 | \Theta = \theta_2) = \bar{\beta}, \end{cases}$$

in order to explicit our knowledge, or lack of, with numeric values and relations.

Now we are able to define the **joint distribution** of the boolean random variables A, B, C :

A, B, C	P	Obs.
$a, \neg b, c$	x	$s_1, \Theta = \theta_1$
$\neg a, b, \neg c$	$\bar{x}\beta$	$s_2, \Theta = \theta_2$
$\neg a, \neg b, c$	$\bar{x}\bar{\beta}$	$s_3, \Theta = \theta_2$
*	0	not stable models

where $x, \beta \in [0, 1]$.

- ① Introduction
- ② Extending Probability to Samples
- ③ Cases & Examples
- ④ Conclusions**

- We can use the basics of probability theory and logic programming to assign explicit *parameterized* probabilities to the (stable) models of a program.
- In the covered cases it was possible to define a (parameterized) *family of joint distributions*.
- How far this approach can cover all the cases on logic programs is (still) an issue *under investigation*.
- However, it is non-restrictive since *no unusual assumptions are made*.

- ① Introduction
- ② Extending Probability to Samples
- ③ Cases & Examples
- ④ Conclusions

- An **atom** is $r(t_1, \dots, t_n)$ where
 - r is a n -ary predicate symbol and each t_i is a constant or a variable.
 - A **ground atom** has no variables; A **literal** is either an atom a or a negated atom $\neg a$.
- An **ASP Program** is a set of **rules** such as
$$h_1 \vee \dots \vee h_m \leftarrow b_1 \wedge \dots \wedge b_n.$$
 - The **head** of this rule is $h_1 \vee \dots \vee h_m$, the **body** is $b_1 \wedge \dots \wedge b_n$ and each b_i is a **subgoal**.
 - Each h_i is a literal, each subgoal b_j is a literal or a literal preceded by \sim and $m + n > 0$.
 - A **propositional program** has no variables.
 - A **non-disjunctive rule** has $m \leq 1$; A **normal rule** has $m = 1$; A **constraint** has $m = 0$; A **fact** is a normal rule with $n = 0$.
- The **Herbrand base** of a program is the set of ground literals that result from combining all the predicates and constants of the program.
 - An **event** is a consistent subset (*i.e.* doesn't contain $\{a, \neg a\}$) of the Herbrand base.
 - Given an event I , a ground literal a is **true**, $I \models a$, if $a \in I$; otherwise the literal is **false**.
 - A ground subgoal, $\sim b$, where b is a ground literal, is **true**, $I \models \sim b$ if $b \notin I$; otherwise, if $b \in I$, it is **false**.