Zugzwang

Stochastic Adventures in Inductive Logic

Francisco Coelho

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

November 17, 2022

Notation and Assumptions

- $\overline{x} = 1 - x$.
- **Probabilistic Atomic Choice (PAC):** $x :: a$ defines $a \vee \neg a$ and probabilities $\mathrm{p}(a) = x, \mathrm{p}(\neg a) = \overline{x}$.
- 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* $\mathcal{E}$:

$$\mathcal{E} = \left\{ z \in \mathcal{Z},\ \forall a \in \mathcal{A} \ \middle|\{a, \neg a\} \cap z\middle| \leq 1 \right\}.$$

- *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, $\mathcal{S} = \mathcal{S}_P$, 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:**

$$\mathrm{p}(\theta) = \prod_{a \in \theta} x \prod_{\neg a \in \theta} \overline{x}. \tag{1}$$

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

Disjuntion 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.* ■
- $\mathrm{p}\big(\{\neg a\}\big) = 0.7$ is straightforward.
- But, no *informed* choice for $x \in [0, 1]$ in

$$\mathrm{p}\big(\{a, b\}\big) = 0.3x,$$
$$\mathrm{p}\big(\{a, c\}\big) = 0.3\overline{x}.$$

Lack of Information & Parametrization

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

$$\mathrm{p}\big(\{a, b\}\big) = 0.3x,$$
$$\mathrm{p}\big(\{a, c\}\big) = 0.3\overline{x}.$$

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

$$\mathrm{p}\big(\{\neg a\} \;\big|\; X = x\big) = 0.7,$$
$$\mathrm{p}\big(\{a, b\} \;\big|\; X = x\big) = 0.3x,$$
$$\mathrm{p}\big(\{a, c\} \;\big|\; X = x\big) = 0.3\overline{x}.$$

- Other uncertainties may lead to further conditions:

$$\mathrm{p}(s \mid X_1 = x_1, \ldots, X_n = x_n)\,.$$

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 explicit express such dependences.
- **For example:** TODO Some key examples.

A *random variable* captures this uncertainty:

$$\begin{aligned} \mathrm{p}\big(\{\neg a\} \bigm| X = x\big) &= 0.7, \\ \mathrm{p}\big(\{a, b\} \bigm| X = x\big) &= 0.3x, \\ \mathrm{p}\big(\{a, c\} \bigm| X = x\big) &= 0.3\overline{x}. \end{aligned}$$

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 $d : \mathcal{S} \to [0, 1]$ 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**.

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*?
   - $\mathrm{p}(x) = 0$ for x *excluded* by the specification, including *inconsistent* samples.
   - $\mathrm{p}(x)$ 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:
$$\mathrm{p}(x \mid C = \theta_x) = d(x)\,. \tag{2}$$
  2. $\langle x| \neq \emptyset \wedge |x\rangle = \emptyset$. Then:
$$\mathrm{p}\big(x \mid C = \theta_s, s \in \langle x|\big) = \prod_{s \in \langle x|} d(s)\,. \tag{3}$$
  3. $\langle x| = \emptyset \wedge |x\rangle \neq \emptyset$. Then:
$$\mathrm{p}\big(x \mid C = \theta_s, s \in |x\rangle\big) = \sum_{s \in |x\rangle} d(s)\,. \tag{4}$$
  4. $\langle x| = \emptyset = |x\rangle$. Then:
$$\mathrm{p}(x) = 0. \tag{5}$$

because stable models are *minimal*.

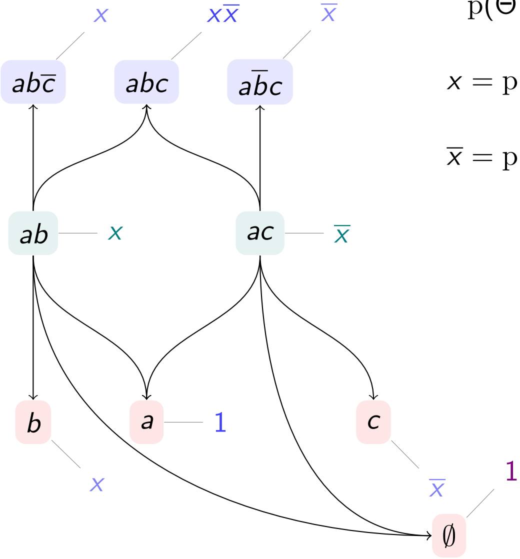
Conditional on Total Choices

- 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:

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

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

Disjunction Example | The Events Lattice



$$\mathrm{p}(E = abc \mid \Theta) = \mathrm{p}(S = ab, S = ac \mid \Theta)$$

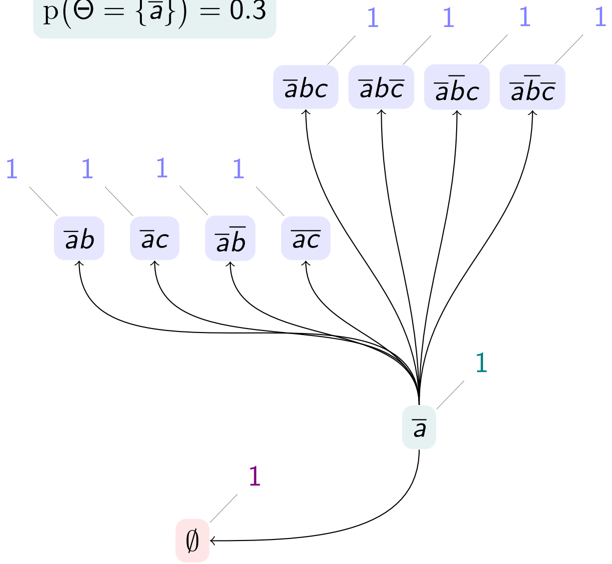
$$\mathrm{p}(\Theta = a) = 0.3$$

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

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

Disjunction Example | The Events Lattice

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



- 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**
$$w_d(x) = 0. \tag{6}$$

2. z is an event and $\langle z| = \{z\} = |z\rangle$. Then z is a stable model and **define**
$$w_d(z) = w(z) = d(z)\,\mathrm{p}(\theta_z)\,. \tag{7}$$

3. z is an event and $\langle z| \neq \emptyset \wedge |x\rangle = \emptyset$. Then **define**
$$w_d(z) = \sum_{s \in \langle z|} w_d(s)\,. \tag{8}$$

4. z is an event and $\langle z| = \emptyset \wedge |z\rangle \neq \emptyset$. Then **define**
$$w_d(z) = \prod_{s \in |z\rangle} w_d(s)\,. \tag{9}$$

5. z is an event and $\langle z| = \emptyset \wedge |z\rangle = \emptyset$. Then **define**
$$w_d(z) = 0. \tag{10}$$

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 $\bot = \emptyset$ 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 $P : F \to [0, 1]$ such that:
   1. $\mathrm{p}(E) \in [0, 1]$ for any $E \in F$.
   2. $\mathrm{p}(\Omega) = 1$.
   3. if $E_1 \cap E_2 = \emptyset$ then $\mathrm{p}(E_1 \cup E_2) = \mathrm{p}(E_1) + \mathrm{p}(E_2)$.
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

$$\mathrm{p}(z) = \frac{w(z)}{}, z \in \Omega \qquad (11)$$

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 $\overline{x} :: \neg a$. This is enough to get $w(s_1) = \overline{x}$ 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:

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

In this diagram:

- Negations are represented as *e.g.* $\overline{a}$ 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.* $\boxed{\overline{a}b}$. The remaining are simply denoted by *e.g.* $a\overline{b}$.
- 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:

$$\begin{aligned} w(abc) &= w(ab)\, w(ac) = x^2 d\overline{d}, \\ w(\overline{a}\cdot\cdot) &= w(\neg a) = \overline{x}, \\ w(a) &= w(ab) + w(ac) = x(d + \overline{d}) = x, \\ w(b) &= w(ab) = dx, \\ w(c) &= w(ac) = \overline{d}x, \\ w(\emptyset) &= w(ab) + w(ac) + w(\neg a) = dx + \overline{d}x + \overline{x} = 1, \\ w\left(a\overline{b}\right) &= 0. \end{aligned}$$

- The total weight is

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**

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

Looking into probabilistic events of the program and/or its models, we define $x = \mathrm{p}(\Theta = \theta_1) \in [0, 1]$ and $\mathrm{p}(\Theta = \theta_2) = \overline{x}$.
Since s_1 is the only stable model that results from $\Theta = \theta_1$, it is natural to extend $\mathrm{p}(s_1) = \mathrm{p}(\Theta = \theta_1) = x$. However, there is no clear way to assign $\mathrm{p}(s_2), \mathrm{p}(s_3)$ since *both models result from the single total choice* $\Theta = \theta_2$. Clearly,

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

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

$$\begin{cases} \mathrm{p}\big(s_2 \mid \Theta = \theta_2\big) = & \beta \in [0, 1] \\ \mathrm{p}\big(s_3 \mid \Theta = \theta_2\big) = & \overline{\beta}, \end{cases}$$

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	$s_1, \Theta = \theta_1$
$\neg a, b, \neg c$	$\overline{x}\beta$	$s_2, \Theta = \theta_2$
$\neg a, \neg b, c$	$\overline{x}\overline{\beta}$	$s_3, \Theta = \theta_2$
$*$	0	not stable models

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

- 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*.

- An **atom** is $r(t_1, \ldots t_n)$ 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 $h_1 \vee \cdots \vee h_m \leftarrow b_1 \wedge \cdots \wedge b_n$.
  - The **head** of this rule is $h_1 \vee \cdots \vee h_m$, the **body** is $b_1 \wedge \cdots \wedge b_n$ 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 $b \not\in I$; otherwise, if $b \in I$, it is **false**.