P-log: Probabilistic Reasoning with Answer Set Programming

Pedro Cabalar

Depto. Computación
University of Corunna, SPAIN

February 16, 2016

Outline

1. Introduction

2. P-log

Probabilistic reasoning with ASP

This chapter is based on:

1. Chitta Baral, Michael Gelfond, Nelson Rushton,
   Probabilistic Reasoning with Answer Sets,
   Theory and Practice of Logic Programming 9(1), 57–144, 2009.

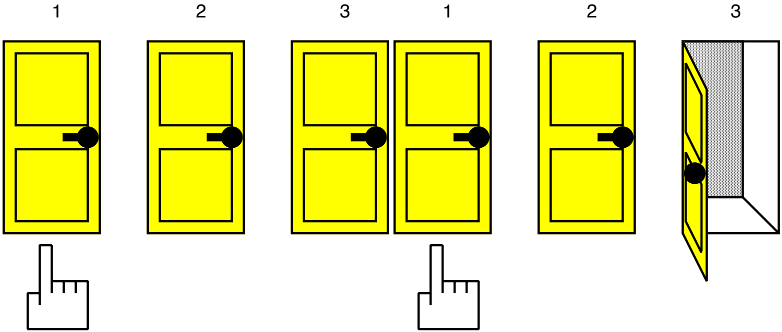
2. Michael Gelfond and Yulia Kahl,
   Knowledge Representation, Reasoning, and the Design of Intelligent Agents,
   Chapter 11, Probabilistic Reasoning.
   (book) (to appear).

Motivation

- Probability Theory: a well studied and developed branch of Mathematics.
- However, its basic notions are not always intuitive for commonsense reasoning.
- This may make Classical Probability Theory alone not suitable for KR. Let us see a pair of examples.

Example 1. Monty Hall problem

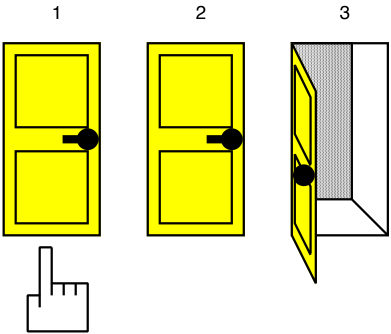
- A player is given the choice to select 1 of 3 closed doors. One of them has a prize and the other two are empty.



- The TV show conductor, Monty, always opens an empty room. Then, he lets the player switch if he likes. Does it really matter?

Example 1. Monty Hall problem

- A player is given the choice to select 1 of 3 closed doors. One of them has a prize and the other two are empty.



- It does: he should switch because he has the double chances!!

Example 2. Simpson's Paradox

Recovery rates for a drug treatment observed among males and females

```
Males:
         fraction_of_population        recovery_rate
 drug             3/8                       60%
-drug             1/8                       70%

Females:
         fraction_of_population        recovery_rate
 drug             1/8                       20%
-drug             3/8                       30%
```

- A patient *P* consults the doctor about trying the drug.

Example 2. Simpson's Paradox

Recovery rates for a drug treatment observed among males and females

```
Males:
        fraction_of_population        recovery_rate
 drug            3/8                       60%
-drug            1/8                       70%

Females:
        fraction_of_population        recovery_rate
 drug            1/8                       20%
-drug            3/8                       30%
```

If *P* is male, the advice is not to take the drug:

$$0,7 = P(recover \mid male, \neg drug) \not< P(recover \mid male, drug) = 0,6$$

Example 2. Simpson's Paradox

Recovery rates for a drug treatment observed among males and females

```
Males:
        fraction_of_population      recovery_rate
 drug            3/8                     60%
-drug            1/8                     70%

Females:
        fraction_of_population      recovery_rate
 drug            1/8                     20%
-drug            3/8                     30%
```

If *P* is female, the advice is not to take the drug either:

$$0.3 = P(recover \mid female, \neg drug) \not< P(recover \mid female, drug) = 0.2$$

Example 2. Simpson's Paradox

Recovery rates for a drug treatment observed among males and females

```
Males:
         fraction_of_population       recovery_rate
 drug              3/8                     60%
-drug              1/8                     70%

Females:
         fraction_of_population       recovery_rate
 drug              1/8                     20%
-drug              3/8                     30%
```

If *P*'s sex is unknown ... the advice is taking the drug ?!?

$$???0.4 \; = P(recover, \neg drug) \; < P(recover, drug) = \; ???0.5$$

Example 2. Simpson's Paradox

- The problem has to do with causal direction between variables.
- Probability Theory cannot tell whether recovery can be an effect of *drug* or vice versa (i.e., when they do not recover, they take the drug, for instance).
- Judea Pearl's causal networks are a variant of Bayesian networks where we can cut a link by causal intervention:

  $$P(recover \mid do(drug))$$

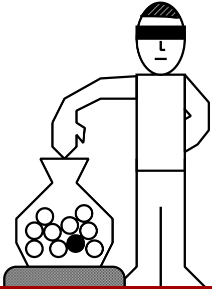
  means probability of recovering when we fix *drug*.

Probabilistic Reasoning

- [Gelfond & Kahl 13] From a KR perspective, probabilistic reasoning =

  commonsense reasoning about the degree of an agent's belief in the likelihood of different events

- An illustrative example: lost in the jungle you are captured by natives that will help you to survive if you (blindly) extract a stone from an urn of the color, black or white, you previously select.



You are told that 9 stones are white and one is black.

Which color should you tell?

Probabilistic models

- Suppose we enumerate the stones $1, \ldots, 10$ and stone 1 is the black one.

A Probabilistic model consists of:

1. $\Omega = \{W_1, W_2, \ldots\}$ possible worlds.
2. $\mu : \Omega \mapsto \mathbb{R}$ probabilistic measure $\mu(W) \geq 0$ agent's degree of belief in likelihood of W.

$$\sum_{W \in \Omega} \mu(\Omega) = 1$$

3. $P : 2^\Omega \mapsto [0, 1]$ probability function

$$P(E) = \sum_{W \in E} \mu(W)$$

We can also use a formula F: $P(F) = P(\{W \mid W \models F\})$

Probabilistic models

- In the example $W_1 = \{selcolor = white, draw = 1, \neg help\}$, $W_2 = \{selcolor = white, draw = 2, help\} \ldots$
- $\mu(W_i)$? Principle of indiference:

  under no preference, possible outcomes of a random experiment are equally probable

  Therefore $\mu(W_i) = \frac{1}{10}$ for $i = 1, \ldots 10$
- Suppose we select *white*, then $P(help) = \frac{9}{10}$. If we select *black* instead, we get $P(help) = \frac{1}{10}$.

P-log

- P-log [Baral, Gelfond, Rushton 09]: main idea
  possible worlds = answer sets of a (probabilistic) logic program.
- The syntax is close to ASP. We allow atoms of the form $a(t) = v$ where a is a functional attribute, t a tuple of terms and v a value in the range of $a(t)$.
- We can declare sorts and types for function arguments and range.
- For Boolean attributes we may use `a(t)` and `-a(t)` to stand for $a(t) = true$ and $a(t) = false$, respectively.
- A P-log program includes additional probabilistic constructs we will see next.

The jungle example in P-log

```
% sorts and general variables
stones={1..10}.
colors={black,white}.
boolean={true,false}.
#domain stones(X).

% Setting the color of each stone
color: stone -> colors.
color(1)=black.
color(X)=white :- X<>1.

% Other attributes
selcolor:colors.    % selected color
help:boolean.
```

The jungle example in P-log

```
% Random variable draw = number of the picked stone
draw:stones.
[r] random(draw).

% Representing the tribal laws
help=true :- draw=X, color(X)=C, selcolor=C.
help=false :- draw=X, color(X)=C, selcolor<>C.

% Suppose we chose white
selcolor=white.

% And we ask the probability of getting help
? {help=true}.
```

The jungle example in P-log

- We make the call

  ```
  plog -t jungle.txt
  ```

  and obtain a probability of 0.9.
- Try with `selcolor=black` instead.

Causal probability statements

- The indiference principle has set all outcomes equally probable, but we can fix probabilities.
- Suppose that, when you select color white, stones are introduced in an irregular urn so that, due to the stone shapes, the probability of picking the black stone is $\frac{1}{3}$.
- We add the statement:

  ```
  [r] pr(draw=1|selcolor=white)=1/3.
  ```
- Which is the probability of getting help now when selecting white? and when we select black?
- Compute the probability of picking stone 2 in both cases. When $selcolor = white$ the rest of stones are equally probable $(1 - 1/3)/9 = 2/27 = 0.074$

Observations and interventions

- P-log allows declaring observations as follows:

  `obs(a(t)=v).`

  meaning that we rule out worlds where `a(t)=v` does not hold.
- For a program Π satisfying some reasonable conditions, computing the conditional probability $P(E|a(t) = v)$ is the same than computing $P(E)$ after adding `obs(a(t)=v)` to Π.
- We can also declare interventions as follows:

  `do(a(t)=v).`

  meaning that we fix `a(t)=v` (it becomes a fact) and that attribute `a(t)` is not random any more.

Observations and interventions

- To illustrate the difference, take Simpson's paradox scenario:

```
boolean = {t,f}.
male, recover, drug : boolean.
[r1] random(male).
[r2] random(recover).
[r3] random(drug).

[r1] pr(male = t)=1/2.
[r2] pr(recover = t | male = t,drug = t) =3/5.
[r2] pr(recover = t | male = t, drug = f)=7/10.
[r2] pr(recover = t | male = f, drug = t)=1/5.
[r2] pr(recover = t | male = f, drug = f)=3/10.
[r3] pr(drug = t|male = t)=3/4.
[r3] pr(drug = t|male = f)=1/4.
```

Observations and interventions

- Try, one by one, the following queries:

```
?{recover=t}|do(drug=t).
?{recover=t}|do(drug=f).
?{recover=t}|obs(drug=t).
?{recover=t}|obs(drug=f).
```

- Using causal interventions yields the expected result (we shouldn't take the drug).
- Using just observations leads to (what seemed a) paradox.

The Monty Hall problem in P-log

- The solution to Monty Hall problem is quite simple. It suffices with limiting the random values that Monty can play with.

```
#domain doors(D).
boolean={true, false}.   doors={1,2,3}.
prize,open,selected:doors.

can_open: doors -> boolean.
can_open(D)=false:- selected=D.
can_open(D)=false:- prize=D.
can_open(D)=true:- not can_open(D)=false.

[r1] random (prize).
[r3] random (open:{X:can_open(X)}).
[r2] random (selected).
?{prize=3}|obs(selected=1),obs(open=2).
```