Probabilistic Programming Languages: Independent Choices and Deterministic Systems

David Poole

Department of Computer Science, University of British Columbia

November 2014
the way causal models were first introduced into genetics, econometrics, and the social sciences, as well as the way causal models are used routinely in physics and engineering . . . causal relationships are expressed in the form of deterministic functional equations, and probabilities are introduced through the assumption that certain variables in the equations are unobserved. This reflects Laplace’s (1814) conception of natural phenomena, according to which nature’s laws are deterministic and randomness surfaces owing merely to our ignorance of the underlying boundary conditions. . . .

we shall express preference towards Laplace’s quasi-deterministic conception of causality . . .

—Judea Pearl [2000] page 26
Outline

1. Semantics of Probabilistic Programming Languages
2. Conditioning on Observations
Probabilistic Programming Languages

- Probabilistic inputs (used in Simula in 1966)
- Conditioning on observations, and querying for distributions
- Inference: more efficient than rejection sampling
- Learning probabilities from data
Representing Bayesian networks

begin
    Boolean a, b, c;
    a := draw(0.1);
    if a then
        b := draw(0.8);
    else
        b := draw(0.3);
    end
    if b then
        c := draw(0.4);
    else
        c := draw(0.75);
end

\[
P(a) = 0.1, \\
P(b | a) = 0.8, \\ P(b | \neg a) = 0.3, \\
P(c | b) = 0.4, \\ P(c | \neg b) = 0.75.
\]

\[
b \iff (a \land bifa) \lor (\neg a \land bifna) \\
c \iff (b \land cifb) \lor (\neg b \land cifnbc)
\]
 Choices among alternatives are independent. Program specifies the consequences of choices.

- **Rejection sampling**: probability of a proposition is the proportion of samples that generate that proposition
- **Independent choice**: possible world for each assignment of a value for each alternative; program specifies what is true in each world
- **Program trace semantics**: possible world for each choice encountered in execution path
- **Abductive semantics**: measure over independent choice worlds; only make distinctions needed to answer a query — provides a measure space over the independent choices
Independent Choice Semantics

\[ P(a) = 0.1, P(bifa) = 0.8, P(bifna) = 0.3, \]
\[ P(cifb) = 0.4, P(cifnb) = 0.75. \]
\[ b \iff (a \land bifa) \lor (\neg a \land bifna) \]
\[ c \iff (b \land cifb) \lor (\neg b \land cifnb) \]

<table>
<thead>
<tr>
<th>World</th>
<th>A</th>
<th>Bifa</th>
<th>Bifna</th>
<th>CIFb</th>
<th>CIFnb</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>(w_0)</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>(0.9 \cdot 0.2 \cdot 0.7 \cdot 0.6 \cdot 0.25)</td>
</tr>
<tr>
<td>(w_1)</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>(0.9 \cdot 0.2 \cdot 0.7 \cdot 0.6 \cdot 0.75)</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(w_{30})</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>(0.1 \cdot 0.8 \cdot 0.3 \cdot 0.4 \cdot 0.75)</td>
</tr>
<tr>
<td>(w_{31})</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>(0.1 \cdot 0.8 \cdot 0.3 \cdot 0.4 \cdot 0.75)</td>
</tr>
</tbody>
</table>

David Poole  Probabilistic Programming Languages
**Program Trace Semantics**

\[
P(a) = 0.1, P(bifa) = 0.8, \ P(bifna) = 0.3, \ P(cifb) = 0.4, \ P(cifnb) = 0.75.
\]

\[
b \iff (a \land bifa) \lor (\neg a \land bifna)
\]

\[
c \iff (b \land cifb) \lor (\neg b \land cifnb)
\]

<table>
<thead>
<tr>
<th>World</th>
<th>A</th>
<th>Bifa</th>
<th>Bifna</th>
<th>Cifb</th>
<th>Cifnb</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>(w_0)</td>
<td>false</td>
<td>⊥</td>
<td>false</td>
<td>⊥</td>
<td>false</td>
<td>(0.9 \times 0.7 \times 0.25)</td>
</tr>
<tr>
<td>(w_1)</td>
<td>false</td>
<td>⊥</td>
<td>false</td>
<td>⊥</td>
<td>true</td>
<td>(0.9 \times 0.7 \times 0.75)</td>
</tr>
<tr>
<td>(\ldots)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(w_6)</td>
<td>true</td>
<td>true</td>
<td>⊥</td>
<td>false</td>
<td>⊥</td>
<td>(0.1 \times 0.8 \times 0.6)</td>
</tr>
<tr>
<td>(w_7)</td>
<td>true</td>
<td>true</td>
<td>⊥</td>
<td>true</td>
<td>⊥</td>
<td>(0.1 \times 0.8 \times 0.4)</td>
</tr>
</tbody>
</table>
PPLs Observations

Program Trace Semantics

\[ P(a) = 0.1, P(bifa) = 0.8, P(bifna) = 0.3, \]
\[ P(cifb) = 0.4, P(cifnb) = 0.75. \]
\[ b \iff (a \land bifa) \lor (\neg a \land bifna) \]
\[ c \iff (b \land cifb) \lor (\neg b \land cifnb) \]

<table>
<thead>
<tr>
<th>World</th>
<th>A</th>
<th>Bifa</th>
<th>Bifna</th>
<th>Cifb</th>
<th>Cifnb</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>(w_0)</td>
<td>false</td>
<td>(\bot)</td>
<td>false</td>
<td>(\bot)</td>
<td>false</td>
<td>(0.9 \times 0.7 \times 0.25)</td>
</tr>
<tr>
<td>(w_1)</td>
<td>false</td>
<td>(\bot)</td>
<td>false</td>
<td>(\bot)</td>
<td>true</td>
<td>(0.9 \times 0.7 \times 0.75)</td>
</tr>
<tr>
<td>(\ldots)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(w_6)</td>
<td>true</td>
<td>true</td>
<td>(\bot)</td>
<td>false</td>
<td>(\bot)</td>
<td>(0.1 \times 0.8 \times 0.6)</td>
</tr>
<tr>
<td>(w_7)</td>
<td>true</td>
<td>true</td>
<td>(\bot)</td>
<td>true</td>
<td>(\bot)</td>
<td>(0.1 \times 0.8 \times 0.4)</td>
</tr>
</tbody>
</table>

Abductive semantics — these are sets of independent choice worlds
begin
  Boolean x;
x := draw(0.2);
if x then
  begin
    Boolean y;
y := draw(0.5);
    ...
  end
else
  begin
    Boolean z;
z := draw(0.7);
    ...
  end

- $y$ only defined when $x$ is true.
- $z$ only defined when $x$ is false.
begin
  Boolean x;
  x := draw(0.2);
  if x then
    begin
      Boolean y;
      y := draw(0.5);
      ...
    end
  else
    begin
      Boolean z;
      z := draw(0.7);
      ...
    end

- y only defined when x is true.
- z only defined when x is false.
- Program trace semantics: y and z are not defined in the same possible worlds.
Semantics Example

begin
  Boolean x;
  x := draw(0.2);
  if x then
    begin
      Boolean y;
      y := draw(0.5);
      ...
    end
  else
    begin
      Boolean z;
      z := draw(0.7);
      ...
    end

- \( y \) only defined when \( x \) is true.
- \( z \) only defined when \( x \) is false.
- Program trace semantics: \( y \) and \( z \) are not defined in the same possible worlds.
- Independent choice semantics: the choices are all independent of each other.

Abductive semantics: worlds which only differ by untaken choices are grouped together.

What program transformations are legal?

David Poole
Probabilistic Programming Languages
begin
  Boolean x;
x := draw(0.2);
if x then
  begin
    Boolean y;
y := draw(0.5);
  ...
  end
else
  begin
    Boolean z;
z := draw(0.7);
  ...
end

- $y$ only defined when $x$ is true.
- $z$ only defined when $x$ is false.
- Program trace semantics: $y$ and $z$ are not defined in the same possible worlds.
- Independent choice semantics: the choices are all independent of each other.
- Abductive semantics: worlds which only differ by untaken choices are grouped together.
Semantics Example

begin
  Boolean x;
  x := draw(0.2);
  if x then
    begin
      Boolean y;
      y := draw(0.5);
      ...
    end
  else
    begin
      Boolean z;
      z := draw(0.7);
      ...
    end

- \( y \) only defined when \( x \) is true.
- \( z \) only defined when \( x \) is false.
- Program trace semantics: \( y \) and \( z \) are not defined in the same possible worlds.
- Independent choice semantics: the choices are all independent of each other.
- Abductive semantics: worlds which only differ by untaken choices are grouped together.
- What program transformations are legal?
begin
  Integer i;
  i := 1;
  while (True)
  begin
    Boolean x;
    x := draw(0.01);
    if x then
      return i;
    else
      i := i+1;
  end
end

- What is the expected value of $i$?
begin
  Integer i;
  i := 1;
  while (True)
  begin
    Boolean x;
    x := draw(0.01);
    if x then
      return i;
    else
      i := i+1;
  end
end

- What is the expected value of $i$?
- How many independent choice worlds are there?
Semantics Example

begin
    Integer i;
    i := 1;
    while (True)
    begin
        Boolean x;
        x := draw(0.01);
        if x then
            return i;
        else
            i := i+1;
    end
end

- What is the expected value of $i$?
- How many independent choice worlds are there?
- What is the probability of the most likely one?
Semantics Example

begin
  Integer i;
  i := 1;
  while (True)
    begin
      Boolean x;
      x := draw(0.01);
      if x then
        return i;
      else
        i := i+1;
    end
end

- What is the expected value of \(i\)?
- How many independent choice worlds are there?
- What is the probability of the most likely one?
- program choice semantics: choices not made are undefined
- abductive semantics: worlds that only differ in choices not made are grouped together
Begin
  Boolean x;
  x := draw(0.2);
  if x then
    begin
      Boolean y;
      y := draw(0.5);
      ...
    end
  else
    begin
      Boolean z;
      z := draw(0.7);
      ...
    end

- y is only defined when x is true.
- z is only defined when x is false.
begin
    Boolean x;
x := draw(0.2);
if x then
    begin
        Boolean y;
y := draw(0.5);
        ...
    end
else
    begin
        Boolean z;
z := draw(0.7);
        ...
    end

- $y$ is only defined when $x$ is true.
- $z$ is only defined when $x$ is false.
- In variable elimination, what happens when $x$ is summed out?
begin
  Boolean x;
  x := draw(0.2);
  if x then
    begin
      Boolean y;
      y := draw(0.5);
      ...
    end
  else
    begin
      Boolean z;
      z := draw(0.7);
      ...
    end
end

- y is only defined when $x$ is true.
- z is only defined when $x$ is false.
- In variable elimination, what happens when $x$ is summed out?
- In MCMC, what happens when $x$ has its value changed?
Boolean x;
x := draw(0.2);
if x then
    return 1;
else
    begin
        x := draw(0.5);
        if x then
            return 2;
        else
            return 3;
    end

What is the probability 1 is returned?
What is the probability 1 is returned?
Boolean x;
x := draw(0.2);
if x then
    return 1;
else
    begin
        x := draw(0.5);
        if x then
            return 2;
        else
            while (True)
                begin
                    x := draw(0.3)
                end
            return 3;
    end

What is the probability 1 is returned?
Boolean x;
x := draw(0.2);
if x then
    return 1;
else
    begin
        x := draw(0.5);
        if x then
            return 2;
        else
            while (True)
                begin
                    x := draw(0.3)
                end
            return 1;
        end
    end

What is the probability 1 is returned?
Boolean x;
x := draw(0.2);
if x then
    return 1;
else
    begin
        x := draw(0.5);
        if x then
            return 2;
        else
            if p_equals_np() then
                return 3;
            else
                return 4;
    end

What is the probability 1 is returned?
Boolean \( x \);
\( x := \text{draw}(0.2) \);
if \( x \) then
    return 1;
else
    begin
        \( x := \text{draw}(0.5) \);
        if \( x \) then
            return 2;
        else
            if \( \text{p.equals_np}() \) then
                return 1;
            else
                return 2;
    end
end

What is the probability 1 is returned?
Outline

1. Semantics of Probabilistic Programming Languages

2. Conditioning on Observations
Observing

- What happens when the vocabulary used in models does not match the vocabulary of observations?
- How can we specify the observations so they interact with programs?
- What happens when observational data and models are build by diverse sets of people?
Given a model of rooms of houses and their colours:

A person observes a house and reports: “The house has a green kitchen.”

What is the probability of the observation?
Probability of an observation

- Given a model of rooms of houses and their colours:
  - A person observes a house and reports: “The house has a green kitchen.”
- What is the probability of the observation?
  - They picked a room at random and reported its colour.
Probability of an observation

- Given a model of rooms of houses and their colours:
  - A person observes a house and reports: “The house has a green kitchen.”

- What is the probability of the observation?
- Why did they tell us this?
  - They picked a room at random and reported its colour.
  - They told us the colour of all of the rooms.
Probability of an observation

- Given a model of rooms of houses and their colours:
- A person observes a house and reports: “The house has a green kitchen.”
- What is the probability of the observation?
- Why did they tell us this?
  - They picked a room at random and reported its colour.
  - They told us the colour of all of the rooms.
  - They searched for a room that is green and reported that they found the kitchen was green.
Probability of an observation

- Given a model of rooms of houses and their colours:
- A person observes a house and reports: “The house has a green kitchen.”
- What is the probability of the observation?
- Why did they tell us this?
  - They picked a room at random and reported its colour.
  - They told us the colour of all of the rooms.
  - They searched for a room that is green and reported that they found the kitchen was green.
  - This was the most interesting/unusual aspect of the house.
Probability of an observation

- Given a model of rooms of houses and their colours:
  - A person observes a house and reports: “The house has a green kitchen.”

- What is the probability of the observation?

- Why did they tell us this?
  - They picked a room at random and reported its colour.
  - They told us the colour of all of the rooms.
  - They searched for a room that is green and reported that they found the kitchen was green.
  - This was the most interesting/unusual aspect of the house.
  - They just finished painting the kitchen.
Probability of an observation

- Given a model of rooms of houses and their colours:
- A person observes a house and reports: “The house has a green kitchen.”
- What is the probability of the observation?
- Why did they tell us this?
  - They picked a room at random and reported its colour.
  - They told us the colour of all of the rooms.
  - They searched for a room that is green and reported that they found the kitchen was green.
  - This was the most interesting/unusual aspect of the house.
  - They just finished painting the kitchen.
- The probability depends on the protocol for observations.
Observe a triangle and a circle touching. What is the probability the triangle is green?

\[ P(\text{green}(x) \mid \text{triangle}(x) \land \exists y \, \text{circle}(y) \land \text{touching}(x, y)) \]

The answer depends on how the \( x \) and \( y \) were chosen!
Protocol for Observing

\[ P(\text{green}(x) \mid \triangle(x) \land \exists y \ circle(y) \land \text{touching}(x, y)) \]

\begin{align*}
\text{select}(x) & \quad \text{select}(y) & \quad \text{select}(x, y) \\
\text{select}(y) & \quad \text{select}(x) & \\
3/4 & \quad 2/3 & \quad 4/5
\end{align*}
Given:
- a database of descriptions of apartments and houses available to rent.
- a set of programs that predict what house a person would be happy with. Each specifies $P(person\_likes \mid description)$.

Want:
- for each house determine which person would most likely want it
- for each person determine which house they would be most likely to like.
Hypothesis about what apartment Mary would like.

Whether Mary likes an apartment depends on:
- Whether there is a bedroom for daughter Sam
- Whether Sam’s room is green
- Whether there is a bedroom for Mary
- Whether Mary’s room is large
- Whether they share

...but apartments don’t come labelled with the roles.
Bayesian Belief Network Representation

Which room is Mary's
Which room is Sam's
Mary's room is large
Sam's room is green
Mary likes her room
Sam likes her room
Need to share
Apartment is suitable

How can we condition on the observation of the apartment?
Naive Bayes representation

How do we specify that Mary chooses a room? What about the case where they (have to) share?
How do we specify that Sam and Mary choose one room each, but they can like many rooms?
Conclusion

- Probabilistic programming language: independent probabilistic choices + deterministic programming language (logic programming, ML, Scheme, Java, C, . . . )
- Need observation languages to complement probabilistic programming languages.
- Many challenges:
  - inference
  - learning
  - conditioning on all relevant data (available anywhere in the world)
  - heterogeneous data sets and semantic interoperability
  - heterogeneous probabilistic models (multiple levels of abstraction and detail)
  - probability of identity and existence