Thinking about an uncertain world

Noah D. Goodman
Stanford University
SymSys 1
Oct 20, 2016
Knowledge

- What do you know about the world?
- How can you use knowledge to think?
- Where does knowledge come from?
Programs

- Computer programs are a way of describing knowledge — what to do.

```
1+2+3+4

sum = function(a,b,c,d){
  return a+b+c+d
}
sum(1,2,3,4)

sum = function(in){
  if(in.length===0) return 0
  else return in[0]+sum(in[1…])
}
sum([1,2,3,4])
```
Uncertainty

Why did he yell at me?
He wanted to hurt me.
He thought I was a telemarketer.
Probabilistic programs

Random primitives:

```javascript
var a = flip(0.3)
var b = flip(0.3)
var c = flip(0.3)
return a + b + c
```

<table>
<thead>
<tr>
<th>Probability / Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

\[
P(n) = \binom{3}{n} 0.3^n 0.7^{3-n}
\]

Sampling \(\approx\) Distributions

Goodman, Mansinghka, Roy, Bonawitz, Tenenabum (2008)
Probabilistic programs

Random primitives:

```javascript
var a = flip(0.3)
var b = flip(0.3)
var c = flip(0.3)
return a + b + c
```

Conditional inference:

```javascript
Infer(
  function(){
    var a = flip(0.3)
    var b = flip(0.3)
    var c = flip(0.3)
    condition( a+b == 1)
    return a + b + c})
```

Goodman, Mansinghka, Roy, Bonawitz, Tenenabum (2008)
**Probabilistic programs**

**Conditional inference:**

```javascript
Infer(
    function(){
        var a = flip(0.3)
        var b = flip(0.3)
        var c = flip(0.3)
        condition( a+b == 1)
        return a + b + c}
)=> 1 0 0 1
```

“It is an old maxim of mine that when you have excluded the impossible, whatever remains, however improbable, must be the truth.”
Probabilistic programs

• Probabilistic programs are a way of describing uncertain knowledge — what the world does.

• And a way to describe thinking (inference) using this knowledge.

• There are several different algorithms that can be used to carry out Infer.

• I.e. this is a computational level theory, in Marr’s sense.
WebPPL

webppl is a small but feature-rich probabilistic programming language embedded in Javascript.

Try it out!
go to:  webppl.org
strength = mem( function(person){return gaussian(10,3)} )
lazy = function(person){return flip(0.1)}
pulling = function(person){
    if( lazy(person) ) return strength(person)/2
    else return strength(person)}
total-pulling = function(team){
    return sum(map(pulling, team))}
beat = function(team1,team2){
    return (total-pulling(team1) > total-pulling(team2))}
Reasoning

Infer()
function() {
    strength = mem( function(person){return gaussian(10,3)} )
    lazy = function(person){return flip(0.1)}
    pulling = function(person){
        if( lazy(person) ) return strength(person)/2
        else return strength(person)
    }
    total-pulling = function(team){
        return sum(map(pulling, team))
    }
    beat = function(team1,team2){
        return (total-pulling(team1) > total-pulling(team2))
    }
    condition(beat([“bob”,“mary”], [“tom”,“sue”]) & beat([“bob”,“bev”], [“jane”,“jim”]))
    return strength(“bob”)
}
Experiment 1

Gerstenberg and Goodman (2012)

Based on the above results, how strong do you think player VE is?

very weak  
very strong
Experiment 1

- 20 conditions: 8 single player tournaments, 12 doubles. (Within subjects design.)
Experiment 2

In Game 2, Player LS was lazy!

Based on the available evidence, how strong do you think player TI is?

very weak  [ ]  very strong

OK
Infer()
function() {
  strength = mem(function(person){return gaussian(10,3)})
  lazy = function(person){return flip(0.1)}
  pulling = function(person){
    if(lazy(person)) return strength(person)/2
    else return strength(person)}
  total-pulling = function(team){
    return sum(map(pulling, team))}
  beat = function(team1,team2){
    return (total-pulling(team1) > total-pulling(team2))
  }
  condition(beat(["bob","mary"], ["tom","sue"])
    & lazy("bob"))
  return strength("bob")
}
Experiment 2

The graph shows a strong correlation with a correlation coefficient of $r = 0.97$ and a root mean square error (RMSE) of 0.287.
Bob has a box with two buttons and a light. He presses both buttons, and the light comes on. How does the box work?

- A alone causes C.
- B alone causes C.
- A or B cause C.
- A and B causes C.
- Nothing causes C.

Goodman, Baker, Tenenbaum (2009; in prep.)
Stuhlmüller & Goodman (2013)
Expt 1: social vs physical

You work at a genetically-engineered plants nursery, and one of your coworkers is tending to some almost-dead flowers that you haven’t seen before.

(Social cond.:) Your coworker pours a yellow liquid and a blue liquid on the flowers.

(Physical cond.:) A small earthquake knocks over a yellow liquid and a blue liquid, which pour on the flowers.

By the end of the day, the flowers are growing again. What causes the flowers to grow?

A only _10$
B only _10$
A or B _20$
A & B _40$
neither _5$

• 9 different cover stories, 3 domains.
Expt 1: social vs physical

N=15

Social condition

Expt 1: social vs physical

Mean Bets ($)

Social condition

Physical condition

Mean Bets ($)

A alone causes C.
B alone causes C.
A or B cause C.
A and B causes C.
Nothing causes C.

A B AorB A&B none

ns

A B AorB A&B none

ns
Learning from actions

True causal structure:

A → C

B → C

Events:

Actions:
Causal learning

function(){
    world = uniform(structures)
    action = uniform(actions)
    outcome = world(action)
    condition(
        action == "press A&B"
        & outcome == "light on")
    return world
}

- Predicts weak inferences (confounded evidence).
Learning from actions

True causal structure: 

Beliefs:

Desires:

Events:

Actions:

Decision = (Knowledge assumption)

Inference
Explaining actions

Beliefs: A \rightarrow B \rightarrow C

Desires:

Actions:

Rational action as inference:

\[ \text{decide} = \text{function}(\text{world}, \text{goal}?) \]
\[ \text{sample}() \]
\[ \text{Infer(function())} \]
\[ \text{action} = \text{action-prior}() \]
\[ \text{outcome} = \text{world}() \]
\[ \text{condition}() \]
\[ \text{return action} \]

Causal learning, revised

function()
{
    world = uniform(structures)
    goal? = uniform(goals)
    action = decide(world, goal?)
    outcome = world(action)
    condition(
        action == "press A&B"
        & outcome == "light on"
    )
    return world
}
Expt 1: social vs physical

N=15

Social condition

A alone causes C.
A and B causes C.
A or B cause C.
Nothing causes C.

Mean Bets ($)

Social-causal model

Probability

Physical condition

N=15

A alone causes C.
B alone causes C.
A or B cause C.
A and B causes C.
Nothing causes C.
Reference games

Speaker: Imagine you are talking to someone and want to refer to the middle object. Would you say “blue” or “circle”?

Listener: Someone uses the word “blue” to refer to one of these objects. Which object are they talking about?

Frank and Goodman (2012)
Recursive reasoning

```
var literalListener = function(property){
    Infer(function(){
        var object = refPrior(context)
        condition(object[property])
        return object
    })
}
```
Recursive reasoning

```javascript
var literalListener = function(property) {
    Infer(function() {
        var object = refPrior(context)
        condition(object[property])
        return object
    })
}

var speaker = function(object) {
    Infer(function() {
        var property = propPrior()
        condition(
            object ==
            sample(literalListener(property)))
        return property
    })
}
```
Recursive reasoning

```javascript
var literalListener = function(property){
    Infer(function(){
        var object = refPrior(context)
        condition(object[property])
        return object
    })
}

var speaker = function(object) {
    Infer(function(){
        var property = propPrior()
        condition(
            object ==
        )
        return property
    })
}

var listener = function(property) {
    Infer(function(){
        var object = refPrior(context)
        condition(utterance ==
            sample(speaker(object)))
        return object
    })
}
```
Recursive reasoning

```javascript
var literalListener = function(property) {
    var object = refPrior(context)
    condition(object[property])
    return object
}

var speaker = function(object) {
    var property = propPrior()
    condition(object = sample(literalListener(property)))
    return property
}

var listener = function(property) {
    Infer(function() {
        var object = refPrior(context)
        condition(utterance == sample(speaker(object)))
        return object
    })
}
```

`refPrior(context)` must be measured experimentally...
Look at the following set of objects:

A  B  C

How many square objects are there?  
How many blue objects are there?  

Now imagine someone is talking to you and uses a word you don't know to refer to one of the objects.

Your job is to decide which object he is talking about. Imagine that you have $100. You should divide your money between the possible objects -- the amount of money you bet on each option should correspond to how confident you are that it is correct. **Bets must sum to 100!**

Which object do you think he is talking about?
A:  B:  C:
• Model explains 98% of variance in data.
language understanding

**listener** = function(utterance) {
    Infer(function() {
        var world = worldPrior()
        var S = **speaker**(world)
        condition(utterance == sample(S))
        return world")
    })
}

**speaker** = function(world) {
    Infer(function() {
        var utterance = utterancePrior()
        var L = **literalListener**(utterance)
        condition(sample(L) == world)
        return utterance")
    })
}

**literalListener** = function(utterance) {
    Infer(function() {
        var world = worldPrior()
        condition(meaning(utterance, world))
        return world")
    })
}

The rational speech act (RSA) framework for pragmatic language understanding. (See Frank & Goodman 2012, Goodman & Stuhlmuller 2013, etc)
Scalar implicature

- Meaning is strengthened by discounting states if there was a more specific utterance that could have been used.

Model Experiment

(a) (c)

Total number of objects with property

<table>
<thead>
<tr>
<th>Number</th>
<th>Probability (x100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>40</td>
<td>2</td>
</tr>
<tr>
<td>60</td>
<td>3</td>
</tr>
<tr>
<td>80</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
</tr>
</tbody>
</table>

"Some", access 1

"Some", access 2

"Some", access 3

(b) (d)

Total number of objects with property

<table>
<thead>
<tr>
<th>Number</th>
<th>Probability (x100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>40</td>
<td>2</td>
</tr>
<tr>
<td>60</td>
<td>3</td>
</tr>
<tr>
<td>80</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
</tr>
</tbody>
</table>

"One", access 1

"One", access 2

"One", access 3

"Two", access 2

"Two", access 3

"Three", access 3

Mean bet

Try it out! go to: webppl.org

Figure 2: (a,b) Model prediction for probability of each world state (number of objects with property), varying the word the speaker used and the speaker's perceptual access. The prior is assumed to be binomial with base rate 0.62, and the speaker optimality parameter is set to $a = 3.4$. (c,d) Mean participant bet on each world state, varying the word the speaker used and the speaker's perceptual access. Data have been filtered to include only trials where the participant's bet that the speaker had complete knowledge was greater than 70 in the expected direction. Error bars are standard error of the mean.

In contrast, when the speaker has only partial access the calculation is more complex, involving the inferred belief distribution of the speaker. Comparing across the three panels of Fig. 2a, we see the probability of 3 is much higher when access is 1 or 2 than when it is 3. When access is 1, no implicature is predicted (the probability of 3 is approximately the same as the probability of 2); when access is 2, only a very slight implicature. Overall, we predict that incomplete speaker knowledge can cancel the standard "some but not all" implicature.

The case of numerals ("one", "two", ...) is similar but more subtle. It has been argued that number words have a lower-bound meaning (Horn, 1972) (e.g. "two balls are red" means $M_2$ of the balls are red), and the intuitive, exact, meanings arise as a pragmatic implicature—"one but not two, three, etc." In Fig. 2b we show model predictions based on the number red apples.
The End

For more on probabilistic models of cognition:
probnmods.org/v2

For more on probabilistic programming languages:
dipl.org

Thanks! to students and collaborators, and funding from: ONR, DARPA, McDonnell Foundation, Sloan Foundation.