Church Tutorial
Probabilistic Programming

Programming + probabilistic modeling

Good representation for AI and cognition

Increasing interest over the past 10 years: BLOG, Bugs, PyMC, ProbLog, Church, Stan, Venture...

Check out: http://probabilistic-programming.org/wiki/Home

https://moalquraishi.wordpress.com/2015/03/29/the-state-of-probabilistic-programming/
The Church Language

Probabilistic program based on Scheme (based on Lisp based on the Lambda calculus)

Compositional, code is data

Several inference engines

Under construction! * *

Founding paper: 
Goodman, Mansinghka, Roy, Bonawitz and Tenenbaum, 2008

Check out forestdb.org

Check out Webppl
Objectives for Tutorial

Become familiar with Church syntax

Run ‘forward’ a few models

Get sense of program/distribution equivalence

mem

Query operator and sampling (rejection sampling, mcmc)

Examples:

- Hypothesis-testing through coin-flipping example
- Causal network inference (medical diagnosis, social inference)
- Intuitive physics and intuitive psychology
Prerequisites and Set-Up

Open local installation of Church if you have one
(i.e. open ‘index.html’ under webchurch/online)

OR

Open https://probmods.org/

AND

Open the ‘church tutorial’ document in the shared dropbox

AND

Play a game of Noisy Tomer Says
Getting Started - Church Syntax

Similar to Scheme/Lisp

Based on $\lambda$-calculus, computing by applying functions

Polish notation: ($+ 2 2$) instead of $2 + 2$
Getting Started - Church Syntax

Math and logic: +, *, >, equal?, and, or...

Naming variables: define

Listing things: list

Quoting things: ‘ (← THIS IS NOT DIRT)

If-ing things: (if condition
  expression1
  expression2)
Getting Started - Church Syntax

Functions: lambda

(define function-name
  (lambda (var1 var2 ... ))
  some-computation)

OR

(define (function-name var1 var2 ...)
  some-computation)
Getting Started - Church Syntax

Other useful notions (let, map, fold, case, ...)

See:

https://www.probmods.org/
Objectives for Tutorial

**Become familiar with Church syntax**

Run ‘forward’ a few models

Get sense of program/distribution equivalence

```
mem
```

Query operator and sampling (rejection sampling, mcmc)

**Examples:**

- Hypothesis-testing through coin-flipping example
- Causal network inference (medical diagnosis, social inference)
- Size principle (number game)
Forward sampling

Exchangeable Random Primitives (XRPs)

Distribution vs. Sampling

Examples:

- Coin flipping
- Gaussian samples
- Memoization
Objectives for Tutorial

Become familiar with Church syntax

Run ‘forward’ a few models <- Generative modeling

Get sense of program/distribution equivalence

mem

Query operator and sampling (rejection sampling, mcmc, etc.)

Examples:

Hypothesis-testing through coin-flipping example

Causal network inference (medical diagnosis)

Planning and social reasoning

Intuitive physics
Inference, Sampling and “query”

Sample generative models (‘run forward’)

Inference (‘run backward’)

© Python (Monty) Picture Ltd. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/.
Inference, Conditioning, sampling and “query”

Syntax:

(q
  what-we-know)

“What we know” is the condition

Setting condition=true is simply sampling from the generative model

This procedure defines a distribution
Rejection Query

(rejection-query
  generative-model
  what-we-want-to-know
  what-we-know)
Implementing Rejection Query

1. Run the model forward

2. Check the condition

3. Accept or repeat
Rejection Query

Very general

Very simple

Very terrible

Physics professors shouldn’t teach geometry.

© Saturday Morning Breakfast Cereal. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/.
MH-query

The backbone of inference in Church

\[(\text{mh-query} \quad \text{num-samples lag})\]

\[\text{generative-model} \quad \text{what-we-want-to-know} \quad \text{what-we-know}\]

Random walk in program evaluation space
MH-query

Very general

Some decisions to make

Could take a while

Biased (burn in)
Objectives for Tutorial

Become familiar with Church syntax

Run ‘forward’ a few models <- Generative modeling

Get sense of program/distribution equivalence

mem

Query operator and sampling (rejection sampling, mcmc, etc.)

Examples:

Hypothesis-testing through coin-flipping example

Causal network inference (medical diagnosis)

Intuitive physics

Planning and social reasoning
Example – Coin Flipping

P(H) = 0.5

P(H) = 0.1

Courtesy of xkcd. License CC BY-NC 2.5.
Example – Coin Flipping

Re-implement Josh’s example of the trick coin

New hypothesis: Biased coin

New new hypothesis: Markov coin

Newest hypothesis: Add your own!
Example – Causal Inference

I used to think correlation implied causation.

Then I took a statistics class. Now I don’t.

Sounds like the class helped. Well, maybe.

Courtesy of xkcd. License CC BY-NC 2.5.
Example – Intuitive Physics

Forward Sampling for Prediction

Inference
Example – Intuitive Psychology

Diagram:
- World
  - Belief
  - Desire
    - Principle of Rational Action
    - Action

Illustrations:
- Banana
- Cartoon character
- Apple
Example – Intuitive Psychology
Example – Social Communication

London

Ben

0.3 1.0