

Probabilistic Programming

Lecture #2: WebPPL

Joost-Pieter Katoen

RWTH Lecture Series on Probabilistic Programming 2022-23

Overview

1 WebPPL Origin and Rationale

2 WebPPL examples

- Randomised quicksort
- Some elementary examples
- Random walks
- Birthday paradox
- Duelling cowboys
- Reasoning about reasoning
- Non-terminating programs

Overview

1 WebPPL Origin and Rationale

2 WebPPL examples

- Randomised quicksort
- Some elementary examples
- Random walks
- Birthday paradox
- Duelling cowboys
- Reasoning about reasoning
- Non-terminating programs

Why WebPPL?

WebPPL	other PPLs
<ul style="list-style-type: none"> • written by cognitive scientists for cognitive scientists • goal: building computational models of cognition • extremely flexible • can be slow 	<ul style="list-style-type: none"> • Stan, Anglican, Alchemy, BUGS, Edward, PyMC • goal: building rich models for data analysis • flexible but limited • faster (for the models that can be expressed)

Recent PPLs (Pyro, Gen) have a steeper learning curve, require more thought about inference, but keep the flexibility of WebPPL and work faster

The mind behind WebPPL



Noah Goodman, Stanford University
 Probabilistic Models of Cognition
dippl.org

WebPPL documentation

- ▶ Basic introduction:
<https://www.problang.org/chapters/app-06-intro-to-webppl.html>
- ▶ Somewhat more extensive introduction with syntax etc.:
<https://uol.de/en/lcs/probabilistic-programming/>
- ▶ Detailed description: available on dippl.org
 Noah Goodman and Andreas Stuhlmüller:
[The Design and Implementation of Probabilistic Programming Languages](#), 2016.

WebPPL principles

- ▶ it is a purely **functional** language
 - ▶ no loops, but can create recursive and higher-order functions
- ▶ one can **easily build generative models**
 - ▶ sample from probability distributions, building simple models, recursion
 - ▶ WebPPL is a language to formally describe how the world works
 - ▶ random choices capture our uncertainty or ignorance
 - ▶ the language is universal: it can express any computable process
 - ▶ causal dependence: the program describes what influences what
- ▶ conditioning on variables, expressions, many inference methods
 - ▶ exact (enumeration), rejection sampling, MCMC, Metropolis-Hastings

[Marta Kryven: Mental Models as Probabilistic Programs, 2020]

Overview

- 1 WebPPL Origin and Rationale
- 2 WebPPL examples
 - Randomised quicksort
 - Some elementary examples
 - Random walks
 - Birthday paradox
 - Duelling cowboys
 - Reasoning about reasoning
 - Non-terminating programs

WebPPL examples

`webppl.org`

Next lecture

Tuesday Oct 18, 16:30