

Project report : Probabilistic programming

Emile Trotignon and Simon Coumes

February 17, 2022

1 Discrete inference by enumeration

1.1 Overview

Simon : Draft phase

We build our language in Ocaml with a library offering a recursive structure and then allow the use of syntactic sugar to improve use comfort. For this first approach, we restrain ourselves to discrete distributions with a finite support. All of this is found in the “finite” sublibrary.

1.2 A recursive structure

Our approach is centered on the introduction of a “model” type, which we inductively defined as follows :

****** Logic style recursive definition.

Note that the construction found in our models is existential. **Simon :** Prettyfy

Show snippets of example code. Assure it will be prettier.

1.3 Recursive evaluation

List all cases quickly. Explain main idea of cartesian product. Continuation for sample.

1.4 Syntactic sugar

Yada. Show snippets of example code.

2 Metropolis Hasting

2.1 Writing continuous distributions

A continuous distribution is either a finite distribution or a pair (sample, logpdf). We implement only one mode of inference for these distributions, Metropolis Hasting. This is found in the continuous sublibrary.

2.2 General theory

Simon : Re-discuss this first

2.3 We already have continuations

Because of the way we wrote the previous logic, it can be followed loosely here while still allowing easy access to continuation.

3 The tests

Simon : Not now