Maximum a Posteriori Estimation by Search in Probabilistic Programs

David Tolpin, Frank Wood
University of Oxford

June 12th, 2015

Paper: `http://arxiv.org/abs/1504.06848`
Slides: `http://offtopia.net/bamc-slides.pdf`

Intuition

Probabilistic program:

- A program with random computations.
- Distributions are conditioned by 'observations'.
- Values of certain expressions are 'predicted' — **the output**.

Can be written in any language (extended by `sample` and `observe`).

Example: Model Selection

```
(let [;; Model
      dist (sample (categorical [[normal 1/4] [gamma 1/4]
                                 [uniform-discrete 1/4]
                                 [uniform-continuous 1/4]]))
      a (sample (gamma 1 1))
      b (sample (gamma 1 1))
      d (dist a b)]

  ;; Observations
  (observe d 1)
  (observe d 2)
  (observe d 4)
  (observe d 7)

  ;; Explanation
  (predict :d (type d))
  (predict :a a)
  (predict :b b)))
```

Inference Objective

- Continuously and **infinitely generate a sequence of samples** drawn from the distribution of the output expression — so that someone else puts it in good use (vague but common).

Inference Objective

- Continuously and **infinitely generate a sequence of samples** drawn from the distribution of the output expression — so that someone else puts it in good use (vague but common).
- Approximately **compute integral** of the form

$$\Phi = \int_{-\infty}^{\infty} \varphi(x) p(x) dx$$

Inference Objective

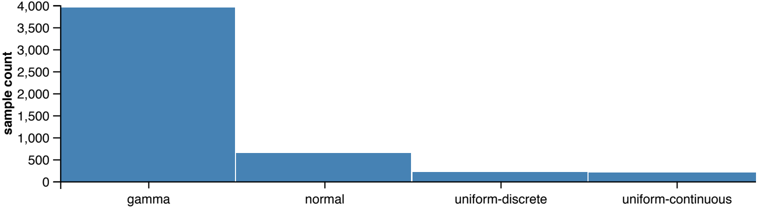
- Continuously and **infinitely generate a sequence of samples** drawn from the distribution of the output expression — so that someone else puts it in good use (vague but common).
- Approximately **compute integral** of the form

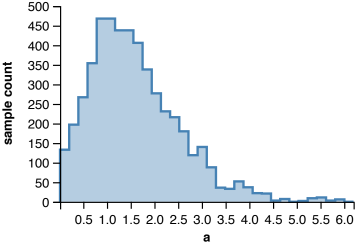
$$\Phi = \int_{-\infty}^{\infty} \varphi(x)p(x)dx$$

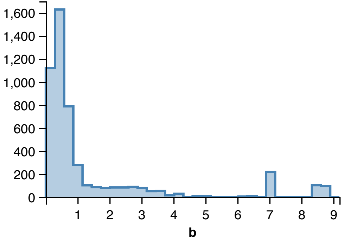
- Suggest **most probable explanation** (MPE) - most likely assignment for all non-evidence variables given evidence. ✓

Example: Inferred Distribution

What we get from probabilistic inference:







Example: Most Probable Explanation

What we (most probably) want to know:

$$Gamma(1.04, 0.28)$$

Monte Carlo Search for MAP

Key features

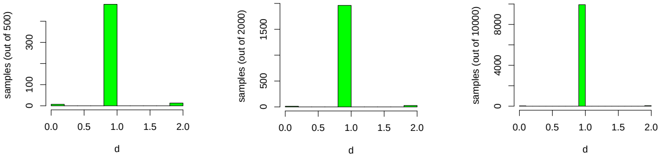
- Inspired by MCTS.
- Maximizes log probability of the program trace.
- Converges much faster than simulated annealing.

Algorithm outline

1. For every random variable, **keep** reward beliefs for seen values.
2. Using *Thompson sampling* (twice):
   - **Guess** reward distribution of a random value.
   - **Select** a value, either seen or randomly drawn.
3. **If** the value is *new*, **add** it to the list of choices.
4. After each run, **back-propagate** log-weight to reward beliefs.

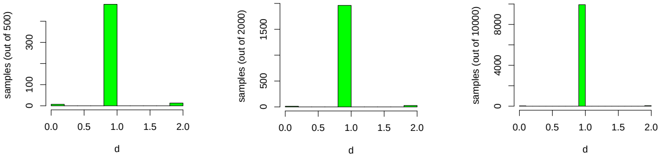
Elementary Distributions: MAP

$$d \sim Discrete(1, 5, 2)$$

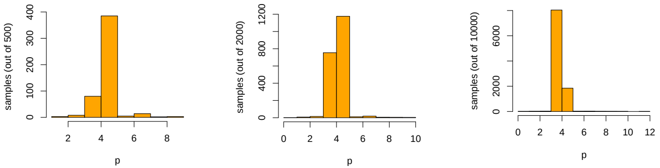


Elementary Distributions: MAP

$$d \sim Discrete(1, 5, 2)$$

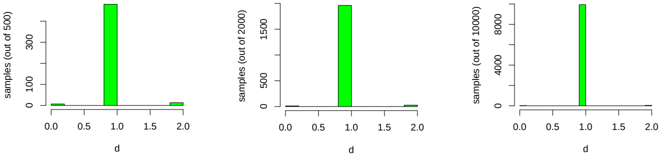


$$p \sim Poisson(5)$$

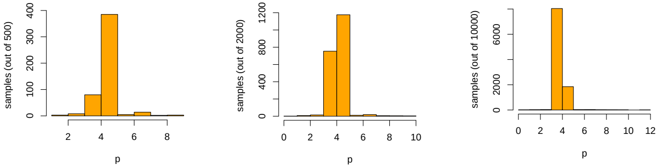


Elementary Distributions: MAP

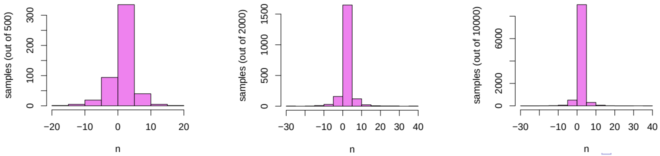
$$d \sim \mathit{Discrete}(1, 5, 2)$$



$$p \sim \mathit{Poisson}(5)$$

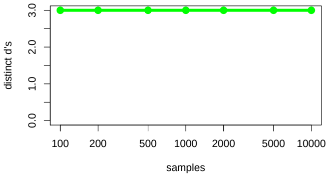


$$n \sim \mathit{Normal}(1, 1)$$

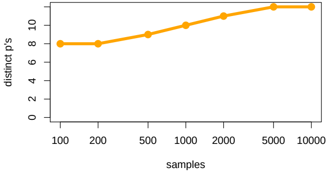


Elementary Distributions: Distinct Values

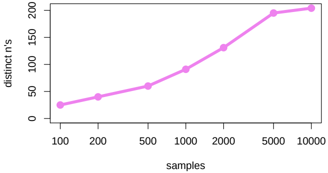
$$d \sim Discrete(1, 5, 2)$$



$$p \sim Poisson(5)$$



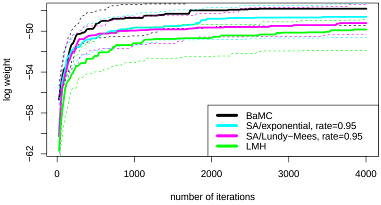
$$n \sim Normal(1, 1)$$



Case Studies

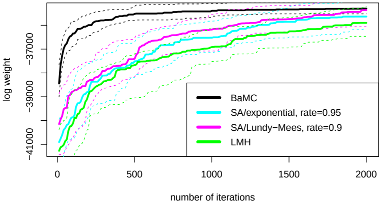
Hidden Markov Model

Dirichlet, discrete, and normal distributions



Probabilistic Deterministic Infinite Automata

Dirichlet and categorical distributions



Summary

- BaMC — a search algorithm for fast MAP estimation.
- Inspired by MCTS.
- Works with any combination of variable types.
- Does not require tuning of parameters.
- Simple to implement.

Thank You