Better Sampling in General Probabilistic Models

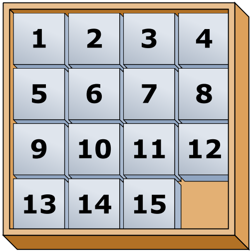
David Tolpin

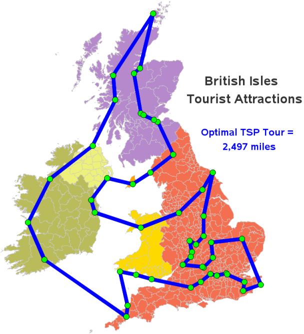
October 1st, 2014

http://offtopia.net/60days-of-research.pdf

[My] Background

- Heuristic Search — get a solution sooner if lucky.
- Metareasoning — how to exploit the luck.





Sampling in Probabilistic Programming

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

```
[assume sigma (sqrt 2)]
[assume mu (normal 1 (sqrt 5))]
[observe (normal mu sigma) 9]
[observe (normal mu sigma) 8]
[predict mu]
```

Sampling Objectives

- Suggest **most probable explanation** (MPE) - most likely assignment for all non-evidence variable given evidence.
- Approximately **compute integral** of the form

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

- 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).

How to sample for an objective?

Deterministic vs. Random Sampling

Random: Draw samples from some [proposal] distribution - simple but

- slow;
- how to sample for an objective?

Deterministic: Select next sample based on the history of earlier samples **and their outcomes**.

Sampling Policies

Goal:

A policy for *online* sample selection.

Tools:

1. Utilities
2. Belief distributions
3. Submodularity of sampling — helps optimize for unknown target.

Case Study: Integration over Probability

Compute an integral:

$$\Phi = \int_{x \in \mathrm{supp}\, p} \varphi(x) p(x) dx \tag{1}$$

Estimate from samples:

$$\hat{\Phi}_K = \frac{\sum_{i=1}^{K} \varphi(x_i)}{K} \tag{2}$$

Importance setting—samples are weighted:

$$\hat{\Phi}_K = \frac{\sum_{i=1}^{k} \varphi(x_i) W_i}{\sum_{i=1}^{K} W_i} \tag{3}$$

Bandit Sampling

Idea:

1. Divide the support into segments, treat each segment as an **arm.**
2. Reward samples by their influence on the estimate: $r = \left|\hat{\Phi}_i - \hat{\Phi}_{i-1}\right|$.
3. Use a bandit algorithm (UCB-something) to select samples.

Problems:

- Arm reward distributions change over time.
- Weights must be properly assigned.
- Integral estimation must be submodular in the samples!

All solvable.

Example: Second Moment of Normal Distribution

Let's estimate:

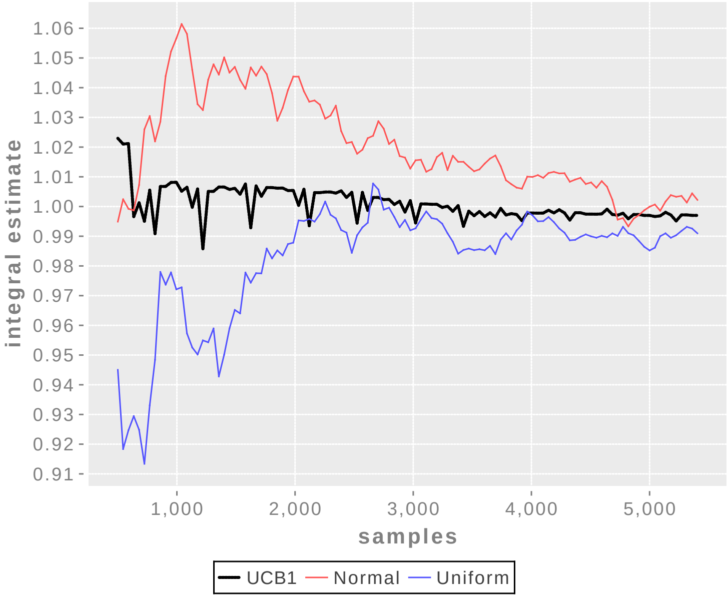
$$\int_{-\infty}^{\infty} x^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \tag{4}$$

We know the answer: **1.**

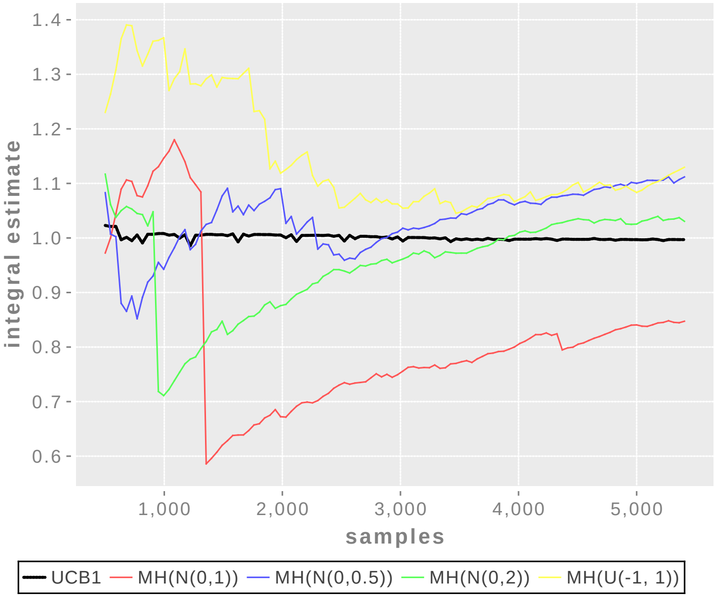
Let's try:

- Importance sampling.
- Metropolis-hastings from $\propto \phi(x)p(x)$.
- Bandit sampling.

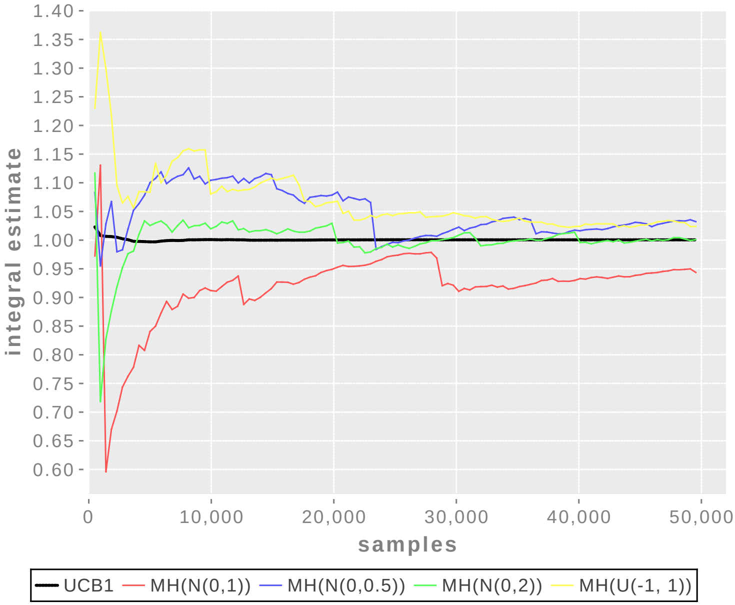
Bandit vs. Non-adaptive Random Sampling



Bandit vs. Metropolis-Hastings - 5,000 samples

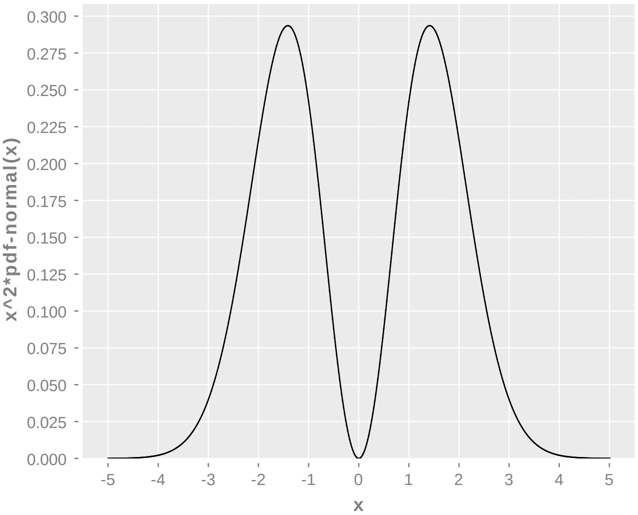


Bandit vs. Metropolis-Hastings - 50,000 samples



Function Shape

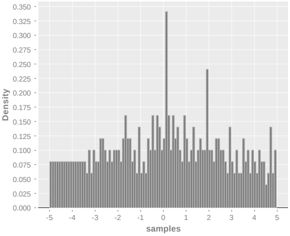
The shape of $\phi(x)p(x)$:

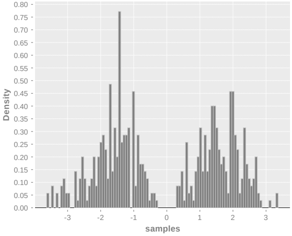


Sample Distribution

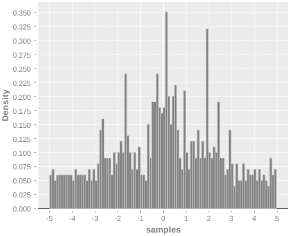
UCB1 | **MH**

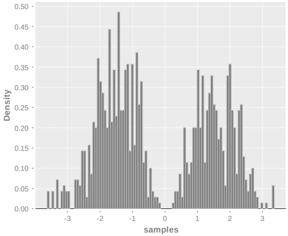
500 samples





1000 samples

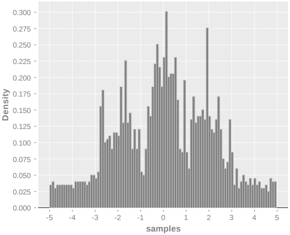


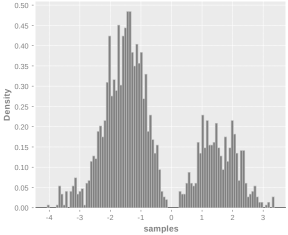


Sample Distribution (cont)

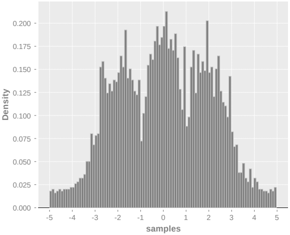
UCB1 **MH**

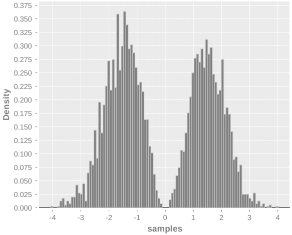
2000 samples





5000 samples

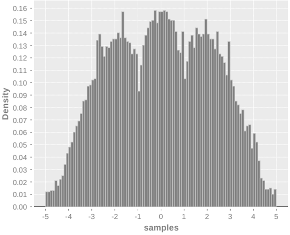


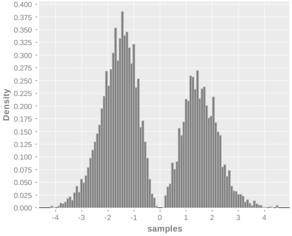


Sample Distribution (cont)

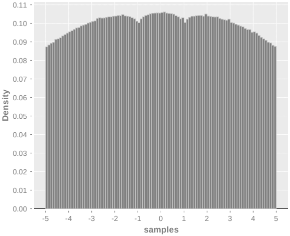
UCB1 **MH**

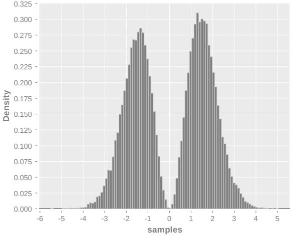
10000 samples





100000 samples





Next Steps

- Bayesian Sampling — Cox Process with Gaussian prior on λ.
- Objective function for ‘just sampling’.
- Better sampling in probabilistic programs.

Thank You