Gov 2002 - Section 3 - Slice Sampling

Connor Jerzak

September 28, 2016

Harvard University, Department of Government

Recap Text Analysis

Warmup Problem!

Slice sampling

Slice sampling, motivated

- We want to obtain samples from $f_X(x)$, where $f_X(x)$ is usually some posterior distribution.
- Metropolis-Hastings methods involve generating candidate values, and accepting or rejecting these values with the appropriate probability.
- Often, it can be hard to ensure candidates are accepted frequently enough, but not too frequently.
- Slice sampling was introduced to avoid having to "reject" any candidate draws.
  - With Metropolis-Hastings, we are essentially generating samples from some distribution by throwing thousands of darts at our distribution, and then discarding all the darts that lie outside our curve. Of course, we will reject many samples this way!

Slice sampling, described

- Introduce a new random variable, U.

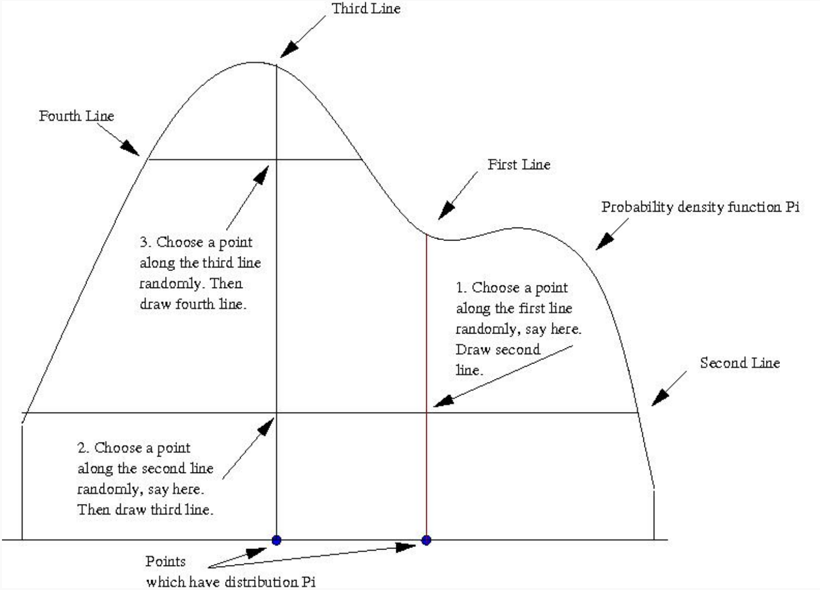
$$X \sim f_x(x);$$
$$U|X \sim \mathrm{Uniform}([0, f_X(x)]).$$

- This gives a joint distribution, $f_{X,U}(x, u)$ where the marginal for X is the original $f_X(x)$.
  - The variable, U, allows us to generate draws from X with the right probability, defined by $f_X(x)$.
    - Sampling over U is equivalent to integrating U out of the joint density, $f_{X,U}(x, u)$ to recover $f_X(x)$.
  - **Intuition**. Sampling from the univariate density, $f_X(x)$ can always be achieved by sampling uniformly from the region under the density $f_X(x)$.

The slice sampling algorithm

For $i = 1, 2, ..., n_{\text{Iterations}}$:

- Sample $u^{(i+1)} \sim \text{Uniform}([0, f_X(x^{(i)})])$
- Sample $x^{(i+1)} \sim \text{Uniform}(\{x : f_X(x) > u^{(i+1)}\})$


Third Line
Fourth Line
First Line
Probability density function Pi
3. Choose a point along the third line randomly. Then draw fourth line.
1. Choose a point along the first line randomly, say here. Draw second line.
Second Line
2. Choose a point along the second line randomly, say here. Then draw third line.
Points which have distribution Pi

Slice sampling, positive features

- Slice sampling algorithms are among the best in terms of convergence speed to the target density. This is partly because all candidate samples are accepted.

Slice sampling, potential problems

- The algorithm requires that we can find a starting point that has positive probability. It can often be non-trivial to find such a starting value.
- It can be difficult to keep track of the slices for x. How do we know when $\{x : f_X(x) > \mu^{(i+1)}\}$? Usually, this involves inverting the probability density function of the target posterior.
- The algorithm requires that we can evaluate the posterior density, $f_X(x)$.
- Some distribution functions may separate into separate communicating islands. The slice sampling algorithm may be harder to implement in these cases.

Acknowledgements

- These slides build from material presented by M. Jordan, Jackman, and others.