An MCMC library for probabilistic programming

Rob Zinkov

June 13th, 2014
Special Thanks to Praveen
Why we need it?

• Prototyping probabilistic programming inference solutions
• Easier exploration of mcmc algorithms
• Easier to combine multiple mcmc strategies
Acceptance ratios tricky to get right

\[ A(x^{(i)}, x^*) = \min \left\{ 1, \frac{p(x^*)q(x^{(i)} | x^*)}{p(x^{(i)})q(x^* | x^{(i)})} \right\} \]
Reversible-jump: trickier still

\[ A_{n \rightarrow m} = \min \left\{ 1, \frac{p(m, x^*_m)}{p(n, x_n)} \times \frac{q(n \mid m)}{q(m \mid n)} \times \frac{q_{m \rightarrow n}(u_{m,n} \mid m, x^*_m)}{q_{n \rightarrow m}(u_{n,m} \mid n, x_n)} \times I_{f_{n \rightarrow m}} \right\} \]
Split-merge proposals

\[
A_{\text{split}} = \min \left\{ 1, \frac{p(k + 1, \mu_{k+1})}{p(k, \mu_k)} \times \frac{1}{k+1} \times \frac{1}{p(u_{n,m})} \times \mathcal{J}_{\text{split}} \right\}
\]

\[
A_{\text{merge}} = \min \left\{ 1, \frac{p(k - 1, \mu_{k-1})}{p(k, \mu_k)} \times \frac{1}{k-1} \times \mathcal{J}_{\text{merge}} \right\}
\]
Caveats

- We are only talking MCMC and no other inference methods
- We will not discuss how to use this library in a probabilistic programming system
Core Primitives
Providing a Density

define a Density type

define data type Probability

define type Target a = T (Density a)
Providing a Proposal Distribution

type Sample a = Rand -> IO a

data Proposal a = P (Density a) (Sample a)
Steps are how we transition from one state to another

\[
\text{type } \text{Step } x = \text{Rand} \rightarrow x \rightarrow \text{IO } x
\]
type Kernel x a = Target a ->
(a -> Proposal a) ->
Step x
Walking the MCMC chain

```
walk :: Step x ->
    x ->
    Int ->
    Rand ->
    Action x a ->
    IO a
```
Demo!
\[ 0.3 \exp\left(-0.2x^2\right) + 0.7 \exp\left(-0.2(x - 10)^2\right) \]
Features we provide

- Blocking proposals
- Cyclic kernels
- Mixture kernels
Further work

- Langevin and Hamiltonian MC
- Approximate MCMC (ABC, Noisy MALA)
- Adaptive MC
- Reversible Jump
Conclusions

Let’s write our inference solutions in more modular ways
Coming very soon to Hackage!
Questions?