Probabilistic Graphical Models
Lecture 17: Markov chain Monte Carlo

Andrew Gordon Wilson

www.cs.cmu.edu/~andrewgw
Carnegie Mellon University

March 18, 2015

Resources and Attribution

Image credits, inspiration, pointers for further reading (SL.):

1. Xing (2014). Markov chain Monte Carlo. Lectures on Probabilistic Graphical Models. (SL. 15)
2. Murray (2009). Markov chain Monte Carlo. Cambridge Machine Learning Summer School. (SL. 7, 11, 12, 13, 16, 19, 20, 24, 36, 37)
3. Murray (2007). Advances in Markov chain Monte Carlo Methods. PhD Thesis. (Detailed descriptions for material in the above reference)
4. Bishop (2006). Pattern Recognition and Machine Learning (PRML). (SL. 8, 14, 22, 31, 35)
5. Geweke (2004). Getting it right: joint distribution tests of posterior simulators, JASA 99(467): 799-804. (SL. 34)
6. MacKay (2003). Information Theory, Inference, and Learning Algorithms. (SL. 25, 43)
7. Rasmussen (2000). The Infinite Gaussian Mixture Model. NIPS. (SL. 27)

Monte Carlo

Monte Carlo approximates expectations with sums formed from sampling.

$$\mathbb{E}[f(x)] = \int f(x)p(x)dx \approx \frac{1}{J}\sum_{j=1}^{J} f(x^{(j)}), \quad x^{(j)} \sim p(x) \tag{1}$$

Monte Carlo

Example: Integrating away the weights in e.g., Bayesian neural network.

- Specify $y(x) = f(x, w)$, for response y and input (predictor) x. Place prior $p(w)$ on the weights w.
- Infer posterior $p(w|\mathcal{D}) \propto \overbrace{p(\mathcal{D}|w)}^{\text{likelihood}} p(w)$ given data $\mathcal{D}$.

Derive the predictive distribution at test input x_*:

$$p(y_*|x_*, \mathcal{D}) = \int p(y_*|w, x_*) p(w|\mathcal{D}) dw \tag{2}$$

$$\approx \frac{1}{J} \sum_{j=1}^{J} p(y_*|w^{(j)}, x_*), \qquad w^{(j)} \sim p(w|\mathcal{D}) \tag{3}$$

But how do we sample from $p(w|\mathcal{D})$?

Monte Carlo Warning!

Marginalisation via prior sampling is dangerous!

$$p(\mathcal{D}) = \int p(\mathcal{D}|w)p(w)dw \tag{4}$$

$$\approx \frac{1}{J}\sum_{j=1}^{J} p(\mathcal{D}|w^{(j)}), \quad w^{(j)} \sim p(w). \tag{5}$$

- Question: do you see the problem?

Monte Carlo Warning!

Marginalisation via prior sampling is dangerous!

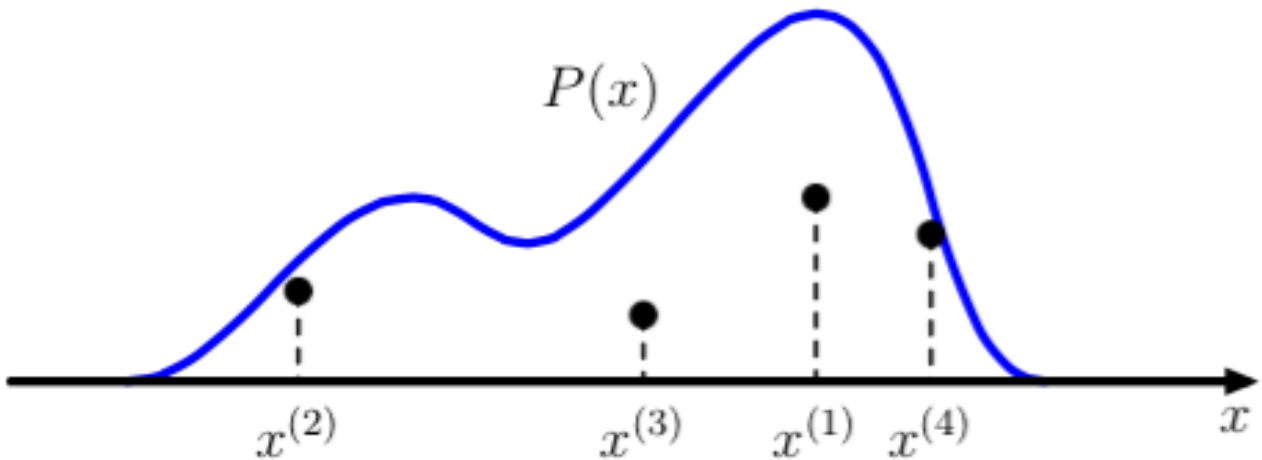
$$p(\mathcal{D}) = \int p(\mathcal{D}|w)p(w)dw \tag{6}$$

$$\approx \frac{1}{J}\sum_{j=1}^{J} p(\mathcal{D}|w^{(j)}), \quad w^{(j)} \sim p(w). \tag{7}$$

- If you were to do multiple finite approximations using this strategy, the variance between the approximations would be massive. Most of the time samples from $p(w)$ will have low likelihood, such that only a small percentage of terms contribute significantly to the Monte Carlo sum.

Monte Carlo

Sampling from $p(x)$ is equivalent to sampling uniformly in the area under $p(x)$.



Suppose that $x \sim U(0,1)$, and we have $y = f(x)$. The distribution of y will be

$$p(y) = p(x)\frac{dx}{dy} = \frac{dx}{dy} \tag{8}$$

Integrating both sides, we find,

$$x = g(y) = \int_{-\infty}^{y} p(y')dy' \tag{9}$$

Therefore $y = g^{-1}(x)$, where g is the CDF of y. To sample from $p(y)$ we can sample from $p(x)$ and then transform the samples with the inverse CDF of y. In other words, sampling uniformly under the curve of $p(y)$ gives samples from y. This is the starting point for many sampling procedures.

Monte Carlo Review

- Monte Carlo: Approximates expectations with sums formed from sampling.
- Variables with uniform distribution under the curve of $p(x)$ are valid samples.
- Cumulative CDF Sampling: If $X \sim U(0,1)$, and $g(\cdot)$ is the CDF of distribution $\mathcal{G}$, then $g^{-1}(X) \sim \mathcal{G}$.

Rejection Sampling

1. Approximate unnormalised $\tilde{p}(x)$ with $kq(x) \geq \tilde{p}(x)$.
2. Sample x_0 from $q(x)$.
3. Sample u_0 from $U(0, kq(x_0))$.
4. x_0, u_0 have uniform distribution under the curve $kq(x_0)$.
5. Accept if $u_0 \leq \tilde{p}(x_0)$.

Importance Sampling

$$\int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx \approx \frac{1}{J}\sum_{j=1}^{J}\frac{p(x^{(j)})}{q(x^{(j)})}f(x^{(j)}),$$

$$x^{(j)} \sim q(x)$$

Review: Sampling from a Bayes Net

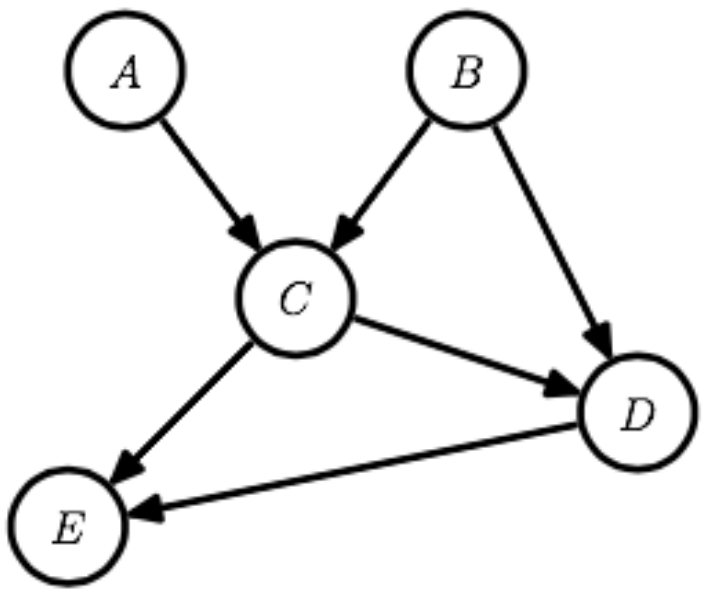
Ancestral Sampling

- Sample top level variables from marginal distributions.
- Sample nodes conditioned on samples of parent nodes.

Example

We wish to sample from
$P(A,B,C,D,E) = P(A)P(B)P(C|A,B)P(D|B,C)P(E|C,D)$.

- $A \sim P(A)$
  $B \sim P(B)$
  $C \sim P(C|A,B)$
  $D \sim P(D|B,C)$
  $E \sim P(E|C,D)$

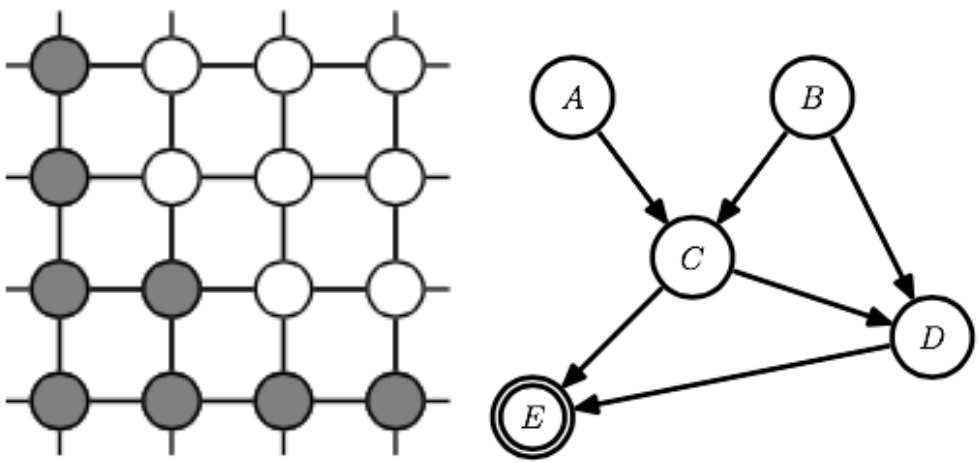


Sampling from High Dimensional Distributions

We often can't decompose $P(x)$ into low dimensional conditional distributions.

- Undirected graphical models: $P(x) = \frac{1}{\mathcal{Z}} \prod_i f_i(x)$.
- Posterior over a directed graphical model: $P(A, B, C, D|E) = P(A, B, C, D, E)/P(E)$.

We often don't know $\mathcal{Z}$ or $P(E)$.



Monte Carlo Limitations: High Dimensional Distributions

Rejection and importance sampling scale badly with dimension. Suppose $p(x) = \mathcal{N}(0, I)$ and $q(x) = \mathcal{N}(0, \sigma^2 I)$.

- We require $\sigma \geq 1$.
  Rejection sampling has an acceptance rate

$$\int \frac{p(x)}{kq(x)} q(x) dx = \frac{1}{k}. \tag{10}$$

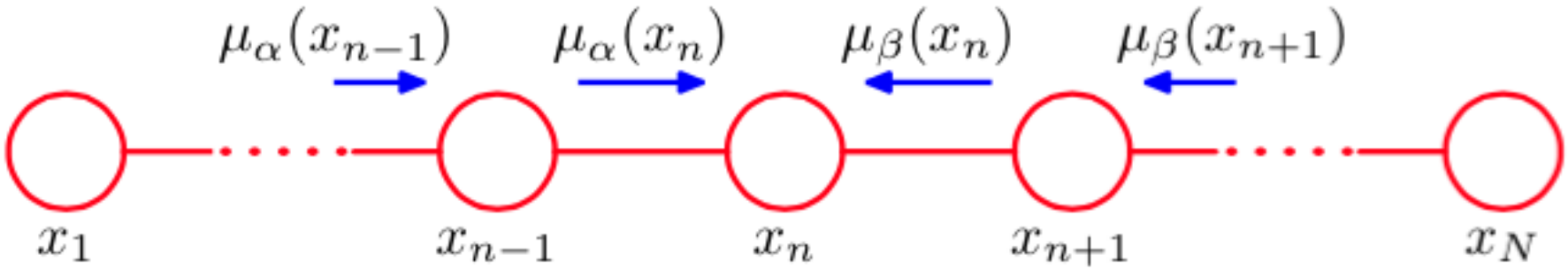
  Here we must set $k = \sigma^{-D/2}$.
- Variance of importance weights is $(\frac{\sigma^2}{2-1/\sigma^2})^{D/2} - 1$.

Generally, for $kq(x) \geq p(x)$, the ratio of the volume outside $p(x)$ to the volume of $p(x)$ shrinks to zero as D increases.

Markov chain Monte Carlo (MCMC)

- Markov chain Monte Carlo methods (MCMC) allow us to sample from a wide array of high dimensional distributions, even when we have very little information about these distributions.

Undirected graphical model for a Markov chain.

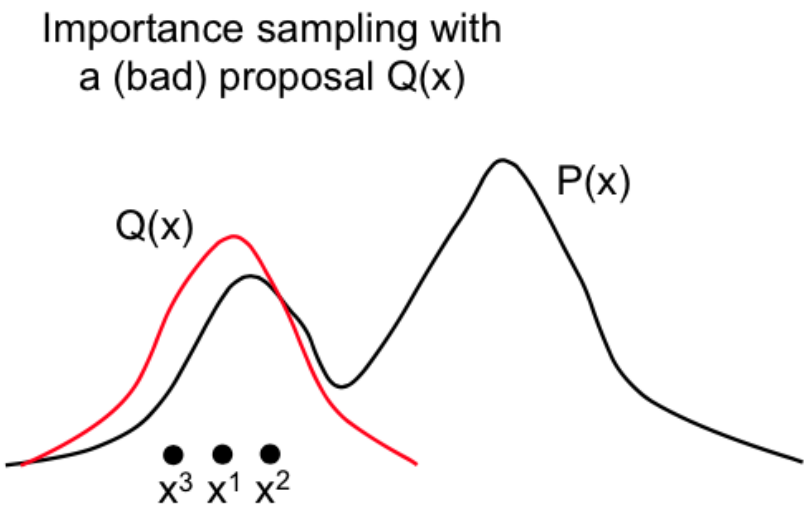


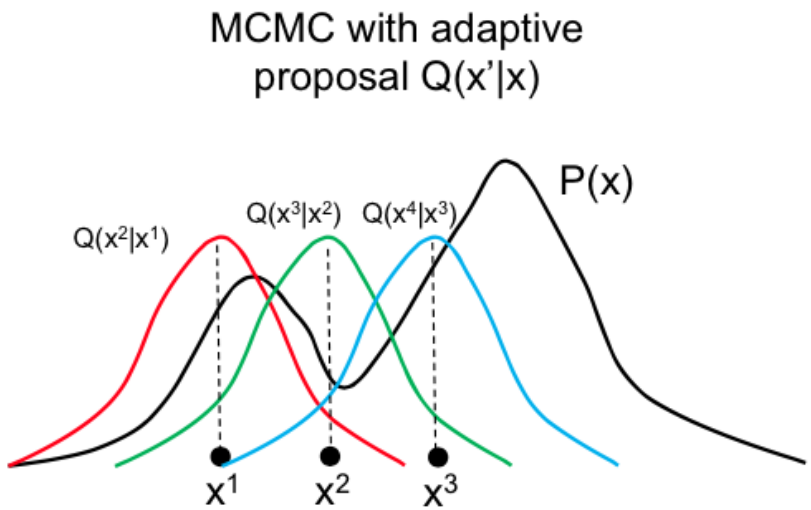
Markov chain Monte Carlo (MCMC)

- MCMC methods allow us to sample from a wide array of high dimensional distributions.
- We sample from a transition probability

$$z_{i+1} \sim T(z_{i+1}|z_i)$$

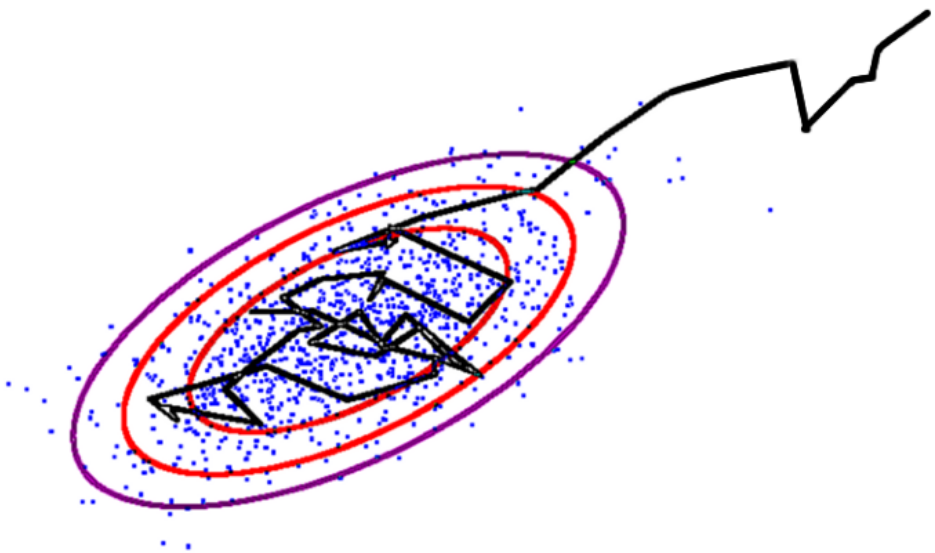
which depends on the current state z_i, an *adaptive* proposal density $q(z_{i+1}; z_i)$, and acceptance rule. Samples $z_1, z_2, \ldots$ therefore form a Markov chain.





Example: Metropolis Algorithm

- Sample proposal x' from a Gaussian distribution $\mathcal{N}(x'; x, \sigma^2)$.
- Accept with probability $\min(1, p(x')/p(x))$.
- If rejected, the next sample is the same as the previous, $x' = x$. This is unlike rejection or importance sampling, where rejected samples are discarded.
- Here we have an adaptive proposal distribution.



Markov chain Monte Carlo (MCMC)

Under what circumstances does the Markov chain converge to the desired distribution? First, some notation and terminology:

- *Transition operator*: $T_i(z_{i+1} \leftarrow z_i) = P(z_{i+1}|z_i)$
- A Markov chain is *homogeneous* if the transition probabilities are the same for all m.
- A distribution $p(z)$ is invariant with respect to a Markov chain if

$$p^*(z) = \sum_{z'} T(z \leftarrow z')p^*(z') \,. \tag{11}$$

- A sufficient *but not necessary* condition for invariant $p(z)$ is to satisfy *detailed balance*:

$$p^*(z')T(z \leftarrow z') = p^*(z)T(z' \leftarrow z) \,. \tag{12}$$

Exercise: prove that detailed balance leads to invariance

Detailed Balance

- A sufficient *but not necessary* condition for invariant $p(z)$ is to satisfy *detailed balance*:

$$p^*(z')T(z \leftarrow z') = p^*(z)T(z' \leftarrow z)\,. \tag{13}$$

What does detailed balance mean?

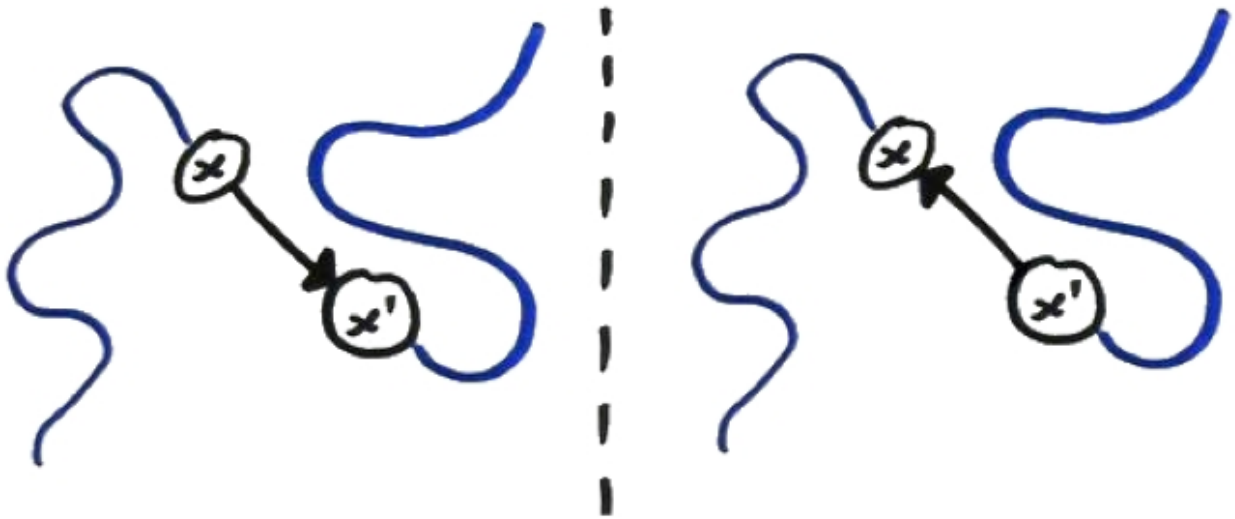
Detailed Balance

- A sufficient *but not necessary* condition for invariant $p(z)$ is to satisfy *detailed balance*:

$$p^*(z')T(z \leftarrow z') = p^*(z)T(z' \leftarrow z)\,. \tag{14}$$

What does detailed balance mean?

It means that $z \leftarrow z'$ and $z' \leftarrow z$ are equally probable.



Reverse Operators

If T is stationary, we can define a reverse operator:

$$\tilde{T}(z \leftarrow z') \propto T(z' \leftarrow z)p^*(z) = \frac{T(z' \leftarrow z)p^*(z)}{\sum_z T(z' \leftarrow z)p^*(z)} = \frac{T(z' \leftarrow z)p^*(z)}{p^*(z')} \tag{15}$$

Generalised Detailed Balance

$$T(z' \leftarrow z)p^*(z) = \tilde{T}(z \leftarrow z')p^*(z') \tag{16}$$

- Generalised detailed balance is both sufficient *and necessary* for invariance.
- Operators satisfying detailed balance are their own reverse operator.

Markov chain Monte Carlo (MCMC)

- Wish to use Markov chains to sample from a given distribution.
- We can do this if
  1. The Markov chain leaves the distribution invariant: $p^*(z) = \sum_{z'} T(z \leftarrow z')p^*(z')$
  2. $\lim_{m\to\infty} p(z_m) = p^*(z)$ regardless of the initial distribution $p(z_0)$ (*ergodicity*).

Creating Transition Operators

Some possibilities:

- Construct transition probabilities from a set of base transitions $B_1, \ldots, B_K$:

$$T(z \leftarrow z') = \sum_{k=1}^{K} \alpha_k B_k(z', z) \tag{17}$$

- Can be combined through successive application:

$$T(z \leftarrow z') = \sum_{z_1} \cdots \sum_{z_{n-1}} B_1(z', z_1) \ldots B_K(z_{K-1}, z) \tag{18}$$

Question: Under what conditions does invariance and detailed balance hold in each case?

Metropolis-Hastings (MH) Algorithm

1. Sample proposal: $x \sim q(x'; x)$; e.g. $\mathcal{N}(x'; x, \sigma^2)$
2. Accept with probability $\min(1, \frac{p(x')q(x;x')}{p(x)q(x';x)})$
3. If rejected, next state in chain is a repeat of the current state (contrast with rejection sampling).

Questions

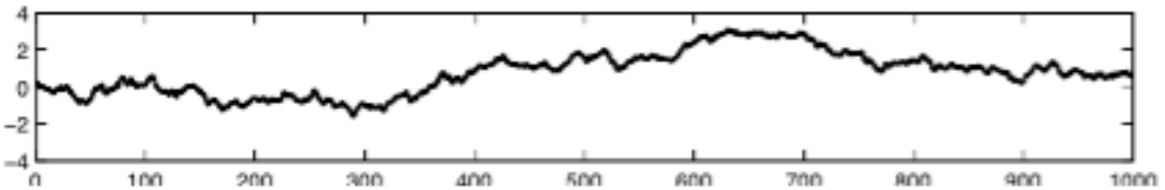
1. Do we require that $p(x)$ be normalised?
2. Does the MH algorithm satisfy detailed balance? Hint: What is the transition operator $T(x' \leftarrow x)$?

MH step-size demo

Exploring $p(x) = \mathcal{N}(x; 0, 1)$ with proposal $q(x'; x) = \mathcal{N}(x'; x, \sigma^2)$ with different step sizes σ:

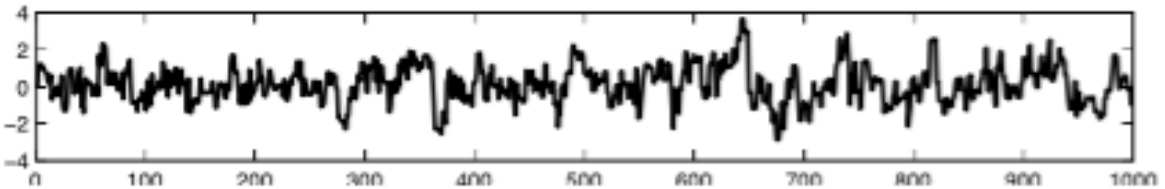
sigma(0.1)

99.8% accepts



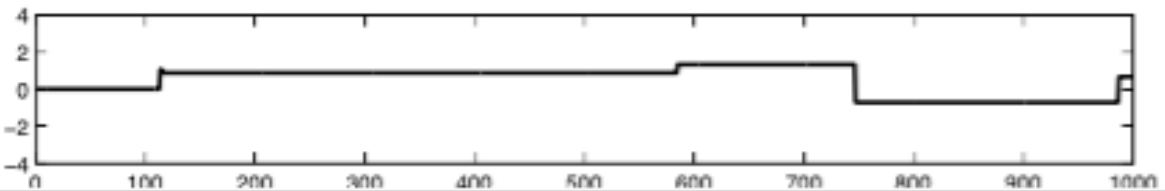
sigma(1)

68.4% accepts



sigma(100)

0.5% accepts



What is the ideal acceptance rate for MH?

- Assume standard Metropolis, with Gaussian $q(x; x')$.
- Accept x' with probability $\lambda = \min(1, p(x')/p(x))$, independent of q.
- All of our information about p is contained from the sequence a of accepts and rejects:
  $a = \{1, 0, 1, 1, 1, 1, 0, 0, 1, \dots\}$.
- a is a sequence of Bernoulli random variables, with parameter λ.
- The entropy (information content) of a is maximized if $\lambda = 0.5$.

MH Summary

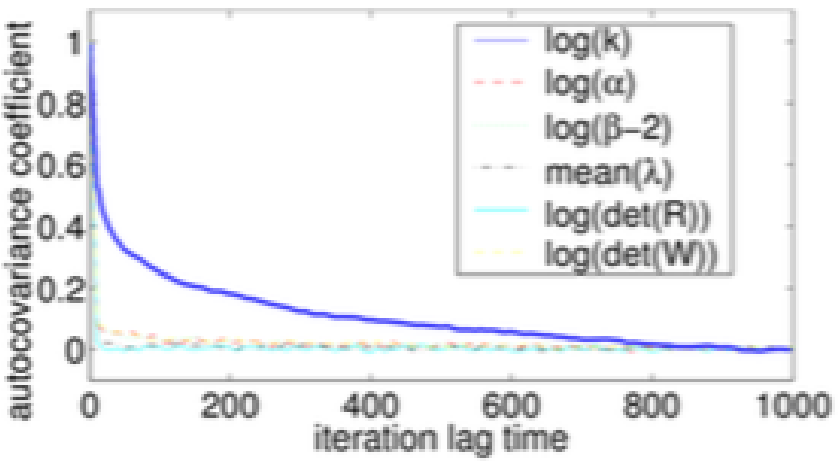
Drawbacks

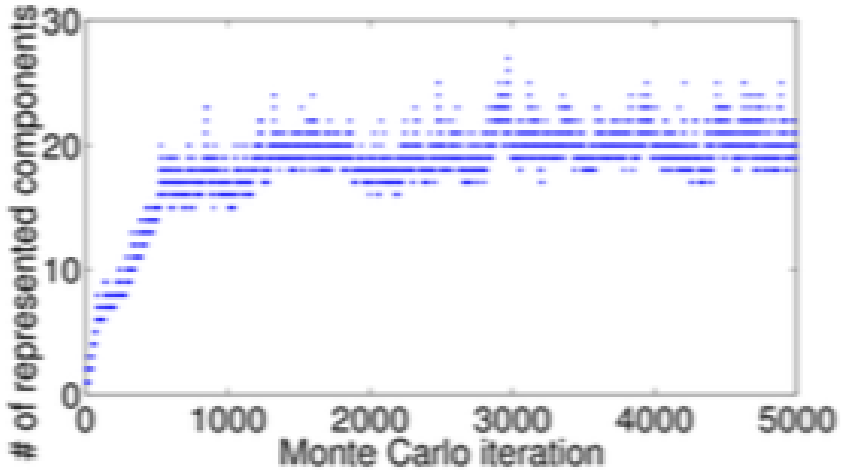
- Large step sizes lead to many rejections
- Small step sizes lead to poor exploration
- Struggles badly with multi-modal distributions (like most popular MCMC strategies)

Benefits

- Simple to implement
- Reasonable for sampling from correlated high dimensional distributions

Assessing convergence





Assessing convergence

- Diagnostics: Plot autocorrelations, compute Gelman-Rubin statistic, packages like R-CODA will tell you the effective number of samples.
- Discussion of thinning, multiple runs, burn in, in *Practical Markov chain Monte Carlo*. Charles J. Geyer, *Statistical Science*. 7(4):473-483, 1992. `http://www.jstor.org/stable/2246094`.
- Unit tests, including running on small-scale versions of your problem, and reasonable inferences on synthetic data drawn from your model, in *Getting it right: joint distribution tests of posterior simulators*, John Geweke, JASA, 99(467): 799-804, 2004.

Auxiliary Variables

Although MCMC is used for marginalisation, sometimes it helps to introduce more variables:

$$\int f(x)p(x)dx = \int f(x)p(x,v)dxdv \approx \frac{1}{J}\sum_{j=1}^{J} f(x^{(j)}), \quad x,v \sim p(x,v)$$

Helps if $p(x|v)$ and $p(v|x)$ are simple, or $p(x,v)$ is easier to navigate than $p(x)$.

Gibbs Sampling

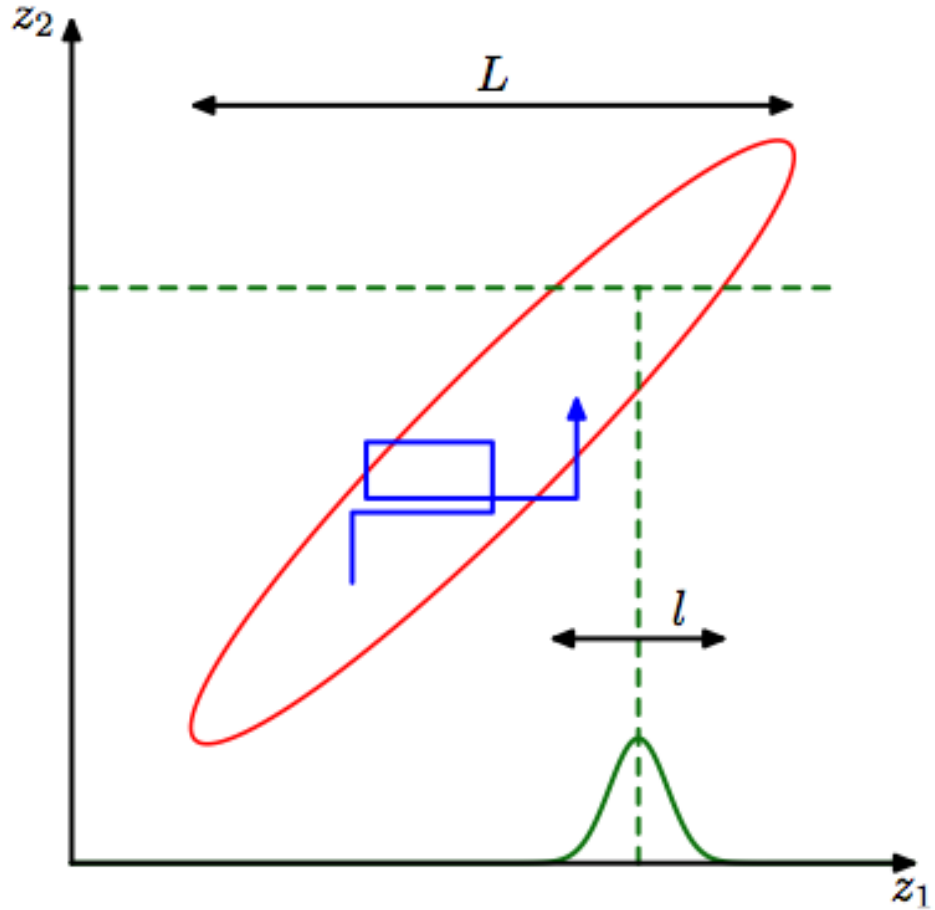
Suppose we wish to sample from the joint distribution $p(A, B, C)$. The algorithm is as follows:

1. Initialize B_0, C_0.
2. Sample $A_0 \sim p(A|B_0, C_0)$
3. $B_1 \sim p(B|A_0, C_0)$
4. $C_1 \sim p(C|A_0, B_1)$
5. Repeat in cycles.

This procedure generalises to arbitrarily many variables.

Gibbs Sampling

We can directly show invariance of the joint distribution and ergodicity under Gibbs sampling. However, Gibbs sampling is a special case of Metropolis-Hastings with proposals $p(x_i|x_{j\neq i})$, which are accepted with probability 1.



Gibbs Sampling

Gibbs sampling is very popular.

Advantages

- Easy access to conditional distributions
- Conditionals may be conjugate (example, Dirichlet process mixtures, next lecture!) and we can sample from them exactly!
- Conditionals will be lower dimensional. We can then apply rejection sampling or importance sampling.
- WinBUGS and OpenBUGS sample from graphical models using Gibbs sampling.
- Can be viewed as a special case of MH with no rejections.

Disadvantages

What might be a drawback?

Collapsed Gibbs Sampling

- Will be discussed in the next lecture.
- Helps overcome some limitations associated with dependencies between variables.
- Is critical for sampling from Dirichlet Process Mixture models (next lecture!)
- Good preparation: C.E. Rasmussen. *The Infinite Gaussian Mixture Model*. NIPS 2000.

Geweke: Getting it right

Consider two routes to sampling from the joint distribution $p(y, w)$ over the data y and parameters w.

1. Sample from your generative model: Sample parameters from your prior w. Sample data given your parameters $p(y|w)$.
2. Sample data from $p(w|y)$ using your transition operator. Resample data $p(y|w)$. This is Gibbs sampling from $p(y, w)$.

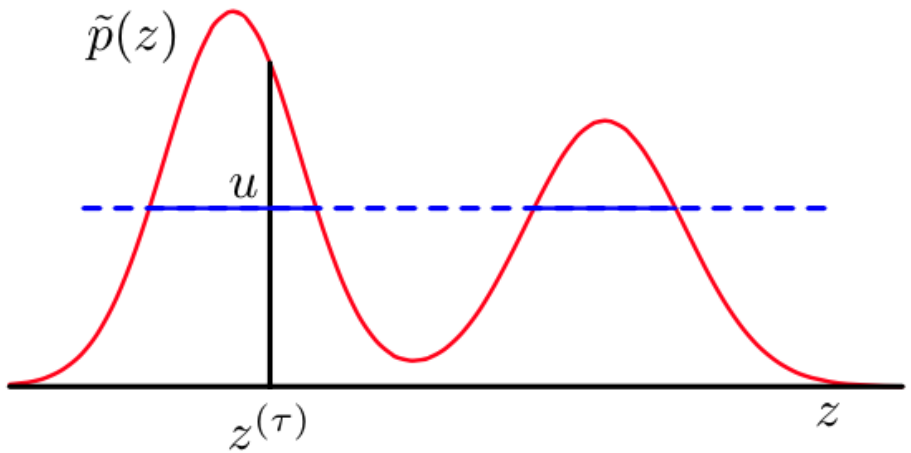
Separately compute the statistics of $p(y, w)$ from each procedure. Discrepancies will be obvious and indicate bugs in your code. Example: suppose there is a bug that increases your posterior noise variance... this will be amplified using these cyclical procedures to sample from $p(y, w)$.

Slice Sampling

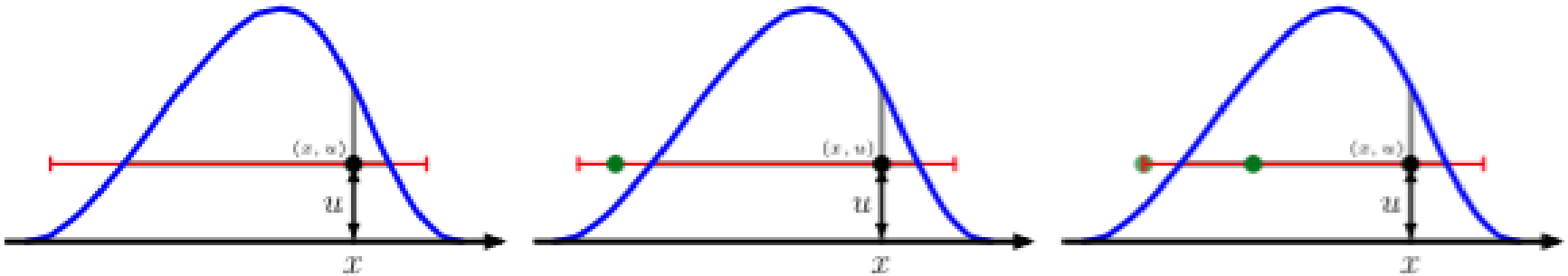
Wish to sample uniformly under the curve $\tilde{p}(z)$.

1. Initialize z_0.
2. Sample $u \sim U(0, \tilde{p}(z))$
3. Sample uniformly from the slice $\{z : \tilde{p}(z) > u\}$. (Step out to find the boundaries).

This procedure samples uniformly under the area of the curve of $\tilde{p}(z)$. It can be viewed as an auxiliary variable MCMC method.

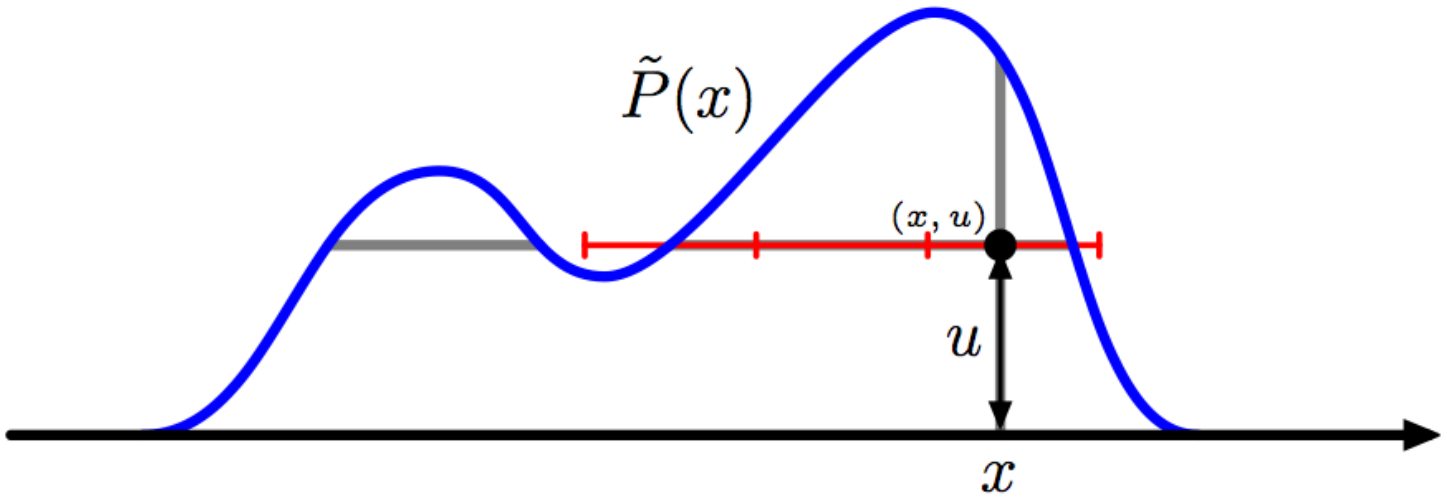


Unimodal Slice Sampling



1. Bracket slice
2. Shrink bracket if uniform sample off slice
3. Keep first sample on slice

Multimodal slice sampling



1. Step out bracket until off slice
2. Sample on slice, shrinking bracket as before.

Slice Sampling: Pros and Cons

Advantages

- Very automatic: lack of tunable free parameters, proposal distributions, etc.
- No rejections.
- Is a great choice when you have little knowledge of the distribution you are sampling from.

Disadvantages

- For multidimensional distributions, one can sample each variable in turn using 1D slice sampling, in a manner analogous to Gibbs sampling. However, this badly suffers from the curse of dimensionality.

Advanced Slice Sampling

If one is sampling from a posterior $p(v|\mathcal{D}) \propto p(\mathcal{D}|v)p(v)$, it is possible to exploit correlations in the prior $p(v)$ for *very efficient* joint updates of v in high dimensional spaces. See, *Elliptical Slice Sampling*. Murray et. al, AISTATS 2009. We will come back to this when we discuss Gaussian processes.

Hamiltonian Monte Carlo

- Often probability distributions can be written in the form $p(x) = \frac{1}{Z} \exp(-E(x))$, where the gradient of $E(x)$ is available.
- The gradient tells us which direction to go to find states of higher probability. It seems wasteful not to use this information!
- Hamiltonian (aka Hybrid) Monte Carlo Methods helps us avoid the basic random walk behaviour of simple Metropolis methods by using gradient information.

Hamiltonian Monte Carlo

- Form $H(x,v) = E(x) + K(v)$, with $K(v) = v^T v/2$.
- $p(x,v) = \frac{1}{\mathcal{Z}_H} \exp[-H(x,v)] = \frac{1}{\mathcal{Z}_H} \exp[-E(x)] \exp[-K(v)]$.
- Since the density is separable, the marginal distribution of x is $p(x)$, so if we can sample from $p(x,v)$ we can simply discard the samples of v for samples of x.
- Simulate Hamiltonian dynamics

Hamiltonian Monte Carlo Algorithm

Benefits

- Very efficient with good settings of τ and ϵ.
- State of the art for sampling from posteriors over Bayesian neural networks.

Drawbacks

- Very difficult to tune τ and ϵ. A recent review and discussion of how to tune HMC is given in R.M. Neal (2011), *MCMC Using Hamiltonian Dynamics*, `http://www.cs.utoronto.ca/~radford/ham-mcmc.abstract.html`.
- HMC helps with *local* exploration, but not with multimodality...

What if we wanted to exploit curvature information too?

Hamiltonian Monte Carlo

```
g = gradE ( x ) ;                  # set gradient using initial x
E = findE ( x ) ;                  # set objective function too

for l = 1:L                        # loop L times
   p = randn ( size(x) ) ;         # initial momentum is Normal(0,1)
   H = p' * p / 2 + E ;            # evaluate H(x,p)

   xnew = x ;  gnew = g ;
   for tau = 1:Tau                 # make Tau 'leapfrog' steps

      p = p - epsilon * gnew / 2 ; # make half-step in p
      xnew = xnew + epsilon * p ;  # make step in x
      gnew = gradE ( xnew ) ;      # find new gradient
      p = p - epsilon * gnew / 2 ; # make half-step in p

   endfor

   Enew = findE ( xnew ) ;         # find new value of H
   Hnew = p' * p / 2 + Enew ;
   dH = Hnew - H ;                 # Decide whether to accept

   if ( dH < 0 )                   accept = 1 ;
   elseif ( rand() < exp(-dH) )    accept = 1 ;
   else                            accept = 0 ;
   endif

   if ( accept )
      g = gnew ;    x = xnew ;     E = Enew ;
   endif
endfor
```

Hamiltonian Monte Carlo

Video

Exploring Multimodal Likelihood Surfaces

Group (blackboard) discussion

- Multiple runs
- Simulated annealing
- Parallel tempering