CSC 412/2506 Spring 2016

Probabilistic Graphical Models

Lecture 7: Sampling

Based on slides by Richard Zemel

Difficult Inference Problems

- Undirected graphical model

- Belief propagation is fast if argument lists of Φ_js are small, and form a junction tree

- For directed graphical models, observations at the leaves induce dependencies amongst latent variables

- One alternative is to utilize a variational approximation, such as mean-field

- Another popular approach: Monte Carlo methods

Monte Carlo Methods

- Useful in many settings, including:
  - Numerical integration
  - Function approximation
  - Optimization

Generally, anywhere we need to compute difficult integrals

- Posterior marginals
- Finding moments (expectations)
- Predictive distributions
- Model comparison

Monte Carlo methods

- Very general approach to Bayesian computation: obtain a sample of points from posterior distribution, use it to make Monte Carlo estimates
- Each sample point will contain values for all the unknown parameters
- Use this sample to approximate expected values by averages over sample points
- Example: sample K values $\theta^1,\ldots,\theta^K$ for a parameter from $P(\theta|\text{data})$, approximate predictive probability:

$$P(y^t = +1 \mid \text{data}) \approx \frac{1}{K}\sum_{j=1}^{K} P(y^t = +1 \mid \theta^j)$$

Simple Monte Carlo

- Statistical sampling can be applied to any expectation
- In general we can find the expectation of *f(x)* by sampling:

$$\int f(x)P(x)dx \approx \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)}) \quad x^{(s)} \sim P(x)$$

- The function *f(x)* is arbitrary, so this is very general

- Example: making predictions

$$P(x|\mathcal{D}) = \int P(x|\theta,\mathcal{D})P(\theta|\mathcal{D})d\theta$$

$$\approx \frac{1}{S}\sum_{s=1}^{S} P(x|\theta^{(s)},\mathcal{D}) \quad \theta^{(s)} \sim P(\theta|\mathcal{D})$$

Properties of Monte Carlo

- Estimator:

$$\int f(x)P(x)dx \approx \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)}) \quad x^{(s)} \sim P(x)$$

- Estimator is unbiased

$$\mathbb{E}_{P(x^{(s)})}[\hat{f}] = \frac{1}{S}\sum_{s=1}^{S}\mathbb{E}_{P(x)}[f(x)] = \mathbb{E}_{P(x)}[f(x)]$$

- Variance shrinks $\propto 1/S$:

$$\mathrm{var}_{P(x^{(s)})}[\hat{f}] = \frac{1}{S^2}\sum_{s=1}^{S}\mathrm{var}_{P(x)}[f(x)] = \mathrm{var}_{P(x)}[f(x)]/S$$

Potential problems with Monte Carlo

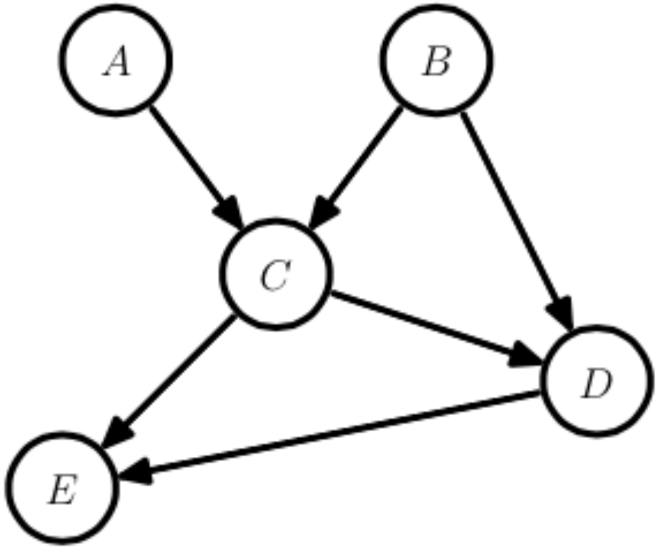
- Samples may not be independent: can have much smaller *effective sample size*

- If computing expectation of f(x), need to ensure that expectation not dominated by regions of low probability

- Next question: how to generate samples?

Sampling from a DGM

- *Ancestral pass* for directed graphical model
  - Sample each top level variable from its marginal
  - Sample each other node from its conditional once its parents have been sampled



Sample:

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

$$P(A, B, C, D, E) = P(A)\, P(B)\, P(C \mid A, B)\, P(D \mid B, C)\, P(E \mid C, D)$$

Sampling from distributions

How to convert samples from a Uniform[0,1] generator?

$$h(y) = \int_{-\infty}^{y} p(y')dy'$$

sample $u \sim$ Uniform[0,1]

$$y(u) = h^{-1}(u)$$

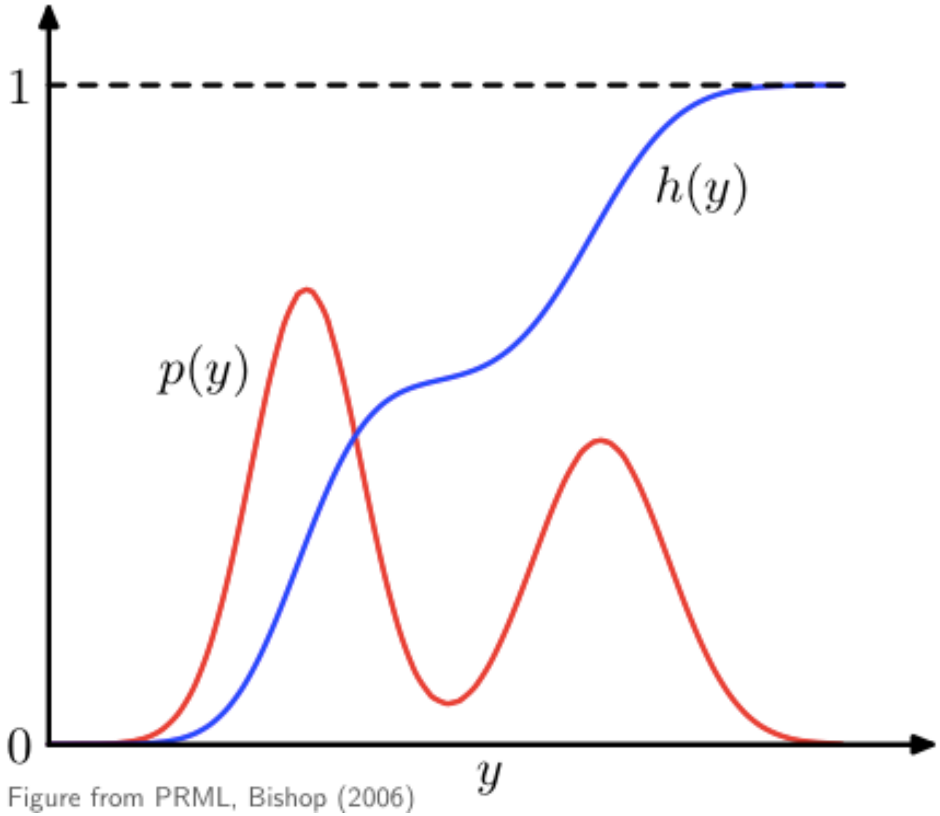


Figure from PRML, Bishop (2006)

But in many cases difficult to compute, invert *h(y)*

Rejection sampling

Can sample *x* non-uniformly: find one that approximates *P(x)*

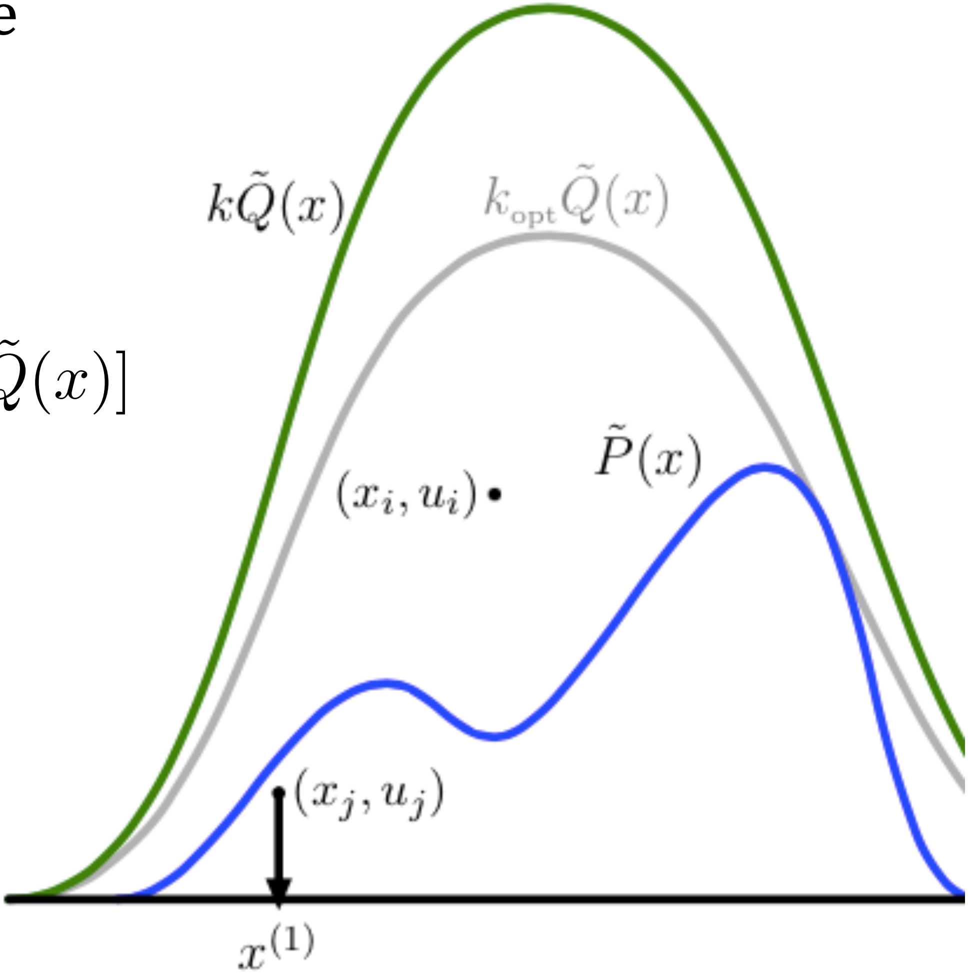
Sampling underneath a $\tilde{P}(x) \propto P(x)$ curve is also valid

Draw underneath a simple curve

$$k\tilde{Q}(x) \geq \tilde{P}(x)$$

- Draw $x \sim Q(x)$
- Height $u \sim \mathrm{Uniform}[0, k\tilde{Q}(x)]$

Discard the point if $u > \tilde{P}(x)$



Importance sampling

Computing both $\tilde{P}(x), \tilde{Q}(x)$ then throwing *x* away seems wasteful

Instead rewrite the integral as an expectation under Q:

$$\int f(x)P(x)dx = \int f(x)\frac{P(x)}{Q(x)}Q(x)dx$$
$$\approx \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)})\frac{P(x^{(s)})}{Q(x^{(s)})}$$

This is just simple Monte Carlo again, so it is unbiased

$r^{(s)} = \dfrac{P(x^{(s)})}{Q(x^{(s)})}$ is called the *importance weight*

Importance sampling also applies when integral is not an expectation

Divide and multiply any integrand by a convenient distribution

Importance sampling (cont.)

Here we assumed we could evaluate $P(x) = \tilde{P}(x)/\mathcal{Z}_P$

If not, we can still apply importance sampling

$$\int f(x)P(x)dx \approx \frac{\mathcal{Z}_Q}{\mathcal{Z}_P}\frac{1}{S}\sum_{s=1}^{S} f(x^{(s)})\frac{\tilde{P}(x^{(s)})}{\tilde{Q}(x^{(s)})}, \quad x^{(s)} \sim Q(x)$$

$$\approx \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)})\frac{\tilde{r}^{(s)}}{\frac{1}{S}\sum_{s'}\tilde{r}^{(s')}} = \sum_{s=1}^{S} f(x^{(s)})w^{(s)}$$

Issues: effective sample size often << S; few samples where *f(x)P(x)* large

Summary

For probabilistic graphical models, often need to compute sums and integrals, eg., expectations, marginal & posterior distributions

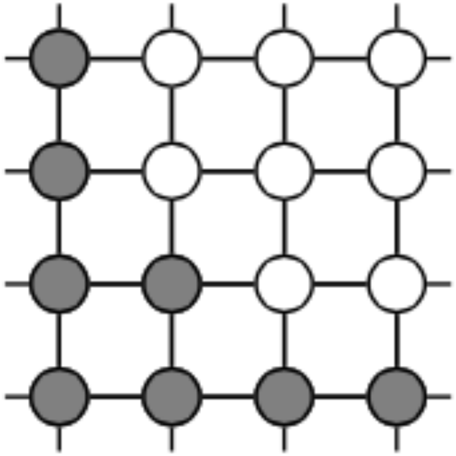
Monte Carlo approximates expectations with a sample average

Rejection sampling draws samples from complex distributions

Importance sampling applies Monte Carlo to sums/integrals in general

Application to large PGMs

But in many probabilistic graphical models, we cannot decompose P(X) into low-dimensional conditionals

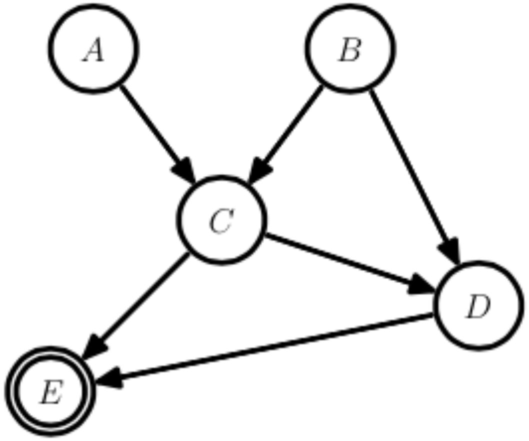


Undirected graphical models:

$$P(x) = \frac{1}{Z}\prod_i f_i(x)$$

Posterior of a directed graphical model

$$P(A,B,C,D|E) = \frac{P(A,B,C,D,E)}{P(E)}$$



But we often don’t know *Z* or *P(E)*

Application to high-dimensional problems

Picking a good sampling distribution becomes hard in high dimensions

Major contributions to integral can be hidden in small areas

Danger of missing those → need to search for high values of *f(x)*

Both rejection and importance sampling scale badly with high D

Example:

$$P(x) = \mathcal{N}(0, I) \qquad Q(x) = \mathcal{N}(0, \sigma^2 I)$$

Rejection: requires σ≥1 → fraction of proposals accepted = σ^{-D}

Importance: variance of importance weights = $(\frac{\sigma^2}{2 - 1/\sigma^2})^{D/2} - 1$

Markov chain Monte Carlo

An alternative is to generate *dependent* samples

Like rejection, importance sampling, here we sample from a proposal distribution

But now maintain record of current state, and proposal distribution depends on it, so sample sequence forms Markov chain

Aim: Design a Markov chain *M* whose moves tend to increase *P(x)* if it is small – the stationary distribution should be the target density

This chain encodes a search strategy: start at an arbitrary *x*, run chain for a while to find an *x* with reasonably high *P(x)*

Picking P_M

MCMC works well if $f(x)/ P_M(x)$ has low variance

$f(x) >> P_M(x)$ means there is a region of comparatively large $f(x)$ that we do not sample enough

$f(x) << P_M(x)$ means that we waste samples in regions where $f(x) \approx 0$

So for example if $f(x) = g(x)\ P(x)$, could ask for $P_M = P$

Metropolis algorithm

- First MCMC algorithm (from 1953), still popular today
- Metropolis algorithm designed to sample from π(x), a Markov chain that transitions from state x to x':
  - Candidate x* is proposed according to some probability S(x,x*)
  - Candidate accepted as next state with probability min[1,π(x*)/ π(x)]

  If x accepted then x'=x*. Else x'=x
- Provided chain can't get trapped, can show that distribution of state gets arbitrarily close to π after sufficiently many transitions
- Good choice of proposal distribution, S, is crucial to getting chain to converge to π rapidly

Metropolis Hastings

Way of getting chain *M* with desired *stationary distribution*

Basic strategy:

- Start from arbitrary *x*
- Repeatedly tweak *x* a little to get *x'*
- If *P (x') ≥ P (x)* move to *x'*
- If *P (x') << P (x)* stay at *x*
- In intermediate cases, randomize

MH has one parameter: how do we tweak *x* to get *x'* ?

Encoded in one-step proposal distribution *Q(x'|x)*

Metropolis Hastings algorithm

MH is defined as follows:

Sample *x' ~ Q(x'/x)*

Compute $$p = \min\left(1, \frac{\tilde{P}(x')Q(x|x')}{\tilde{P}(x)Q(x'|x)}\right)$$

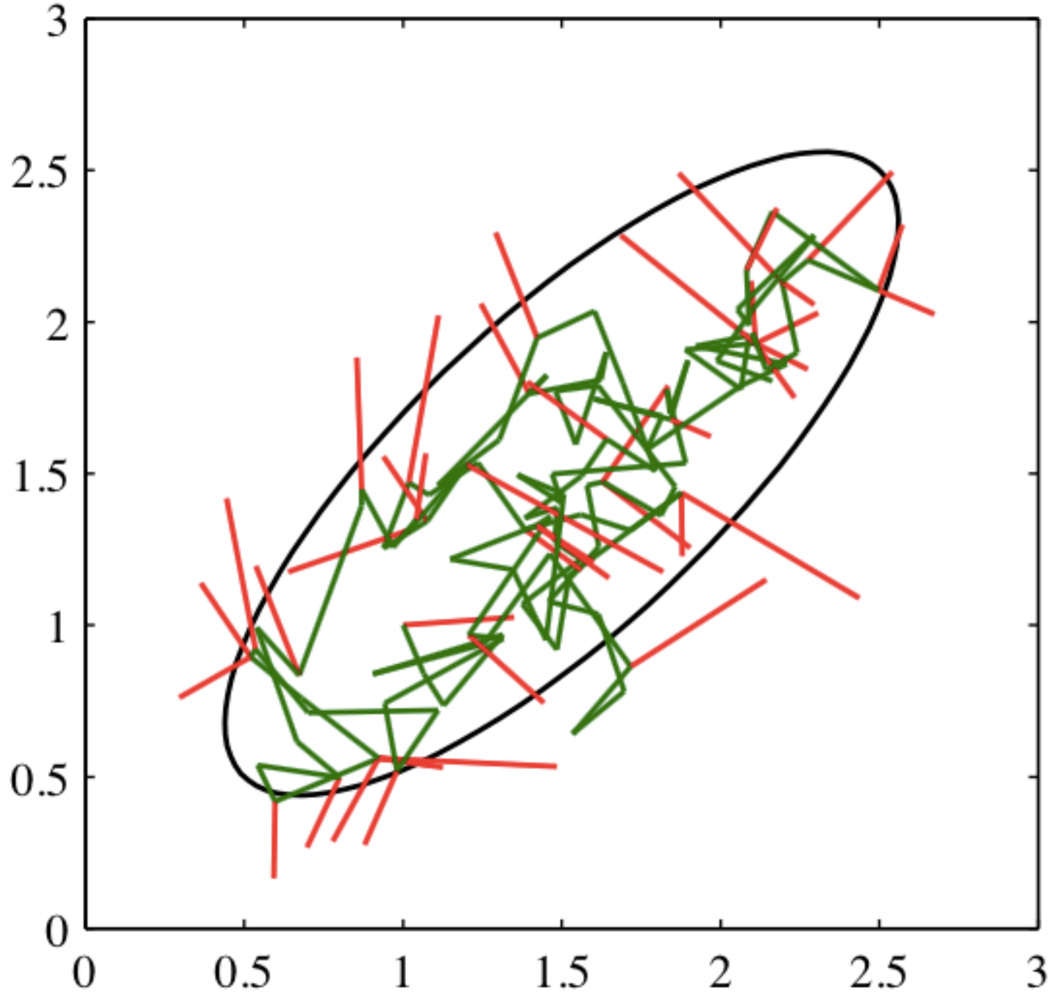
With probability *p*, set *x* ← *x'*

Repeat

Stop after, say, *t* steps (possibly << *t* distinct samples)

Metropolis Hastings algorithm

Example: approximate distribution whose one std deviation contour given by an ellipse, $Q(x'|x), \quad \mathcal{N}(x, \sigma^2)$



Metropolis Hastings notes

Only need *P(x)* up to constant factor: nice for problems where normalizing constant is hard

Efficiency determined by:

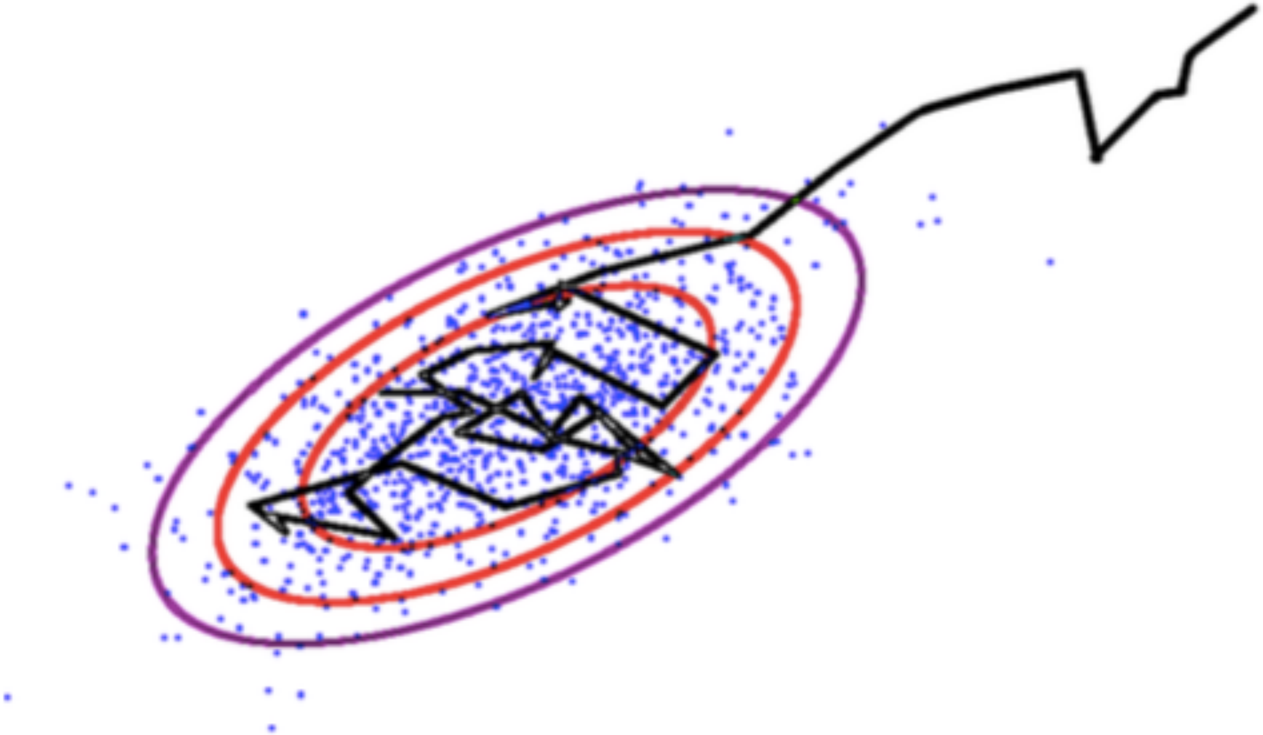
- How fast *Q(x'|x)* moves us around
- How high acceptance probability *p* is

Tension between fast *Q* and high *p*

Metropolis Hastings notes

MCMC gives approximate, correlated samples from the target distribution P*(x)

Uses a biased random walk that explores P*(x)



Valid MCMC operators

Define *transition probabilities* $T(x' \leftarrow x) = P(x'|x)$

Marginals*:* $P(x') = \sum_x P(x'|x)P(x)$

A transition distribution is *invariant*, *or stationary*, wrt a Markov chain if each step leaves that distribution invariant

So the target distribution is invariant if *TP* = P**

$$\sum_x T(x' \leftarrow x)P^*(x) = P^*(x')$$

Also, need to show that distribution converges to required invariant distribution for any initial distribution*: ergodic*

Then *P** is called the *equilibrium distribution*

Discrete example

$$P^\star = \begin{pmatrix} 3/5 \\ 1/5 \\ 1/5 \end{pmatrix} \qquad T = \begin{pmatrix} 2/3 & 1/2 & 1/2 \\ 1/6 & 0 & 1/2 \\ 1/6 & 1/2 & 0 \end{pmatrix} \qquad T_{ij} = T(x_i \leftarrow x_j)$$

*P** is an invariant distribution of *T* because *TP* = P**

$$\sum_x T(x' \leftarrow x) P^*(x) = P^*(x')$$

Also *P** is the equilibrium distribution of *T:*

$$T^{100} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3/5 \\ 1/5 \\ 1/5 \end{pmatrix} = P^\star$$

Ergodicity requires that $T^K(x' \leftarrow x) > 0$ for all *x*': *P*(x')*>0

Detailed balance

Detailed balance means that → *x* → *x'* and → *x'* → x are equally probable

$$T(x' \leftarrow x)\, P^*(x) = T(x \leftarrow x')\, P^*(x')$$

Detailed balance implies the invariant condition

$$\sum_x T(x' \leftarrow x) P^*(x) = \sum_x T(x \leftarrow x') P^*(x') = P^*(x') \sum_x P(x|x') = P^*(x')$$

A Markov chain that respects detailed balance is *reversible*

To show that P is an invariant distribution can show that detailed balance is satisfied*

Metropolis limitations

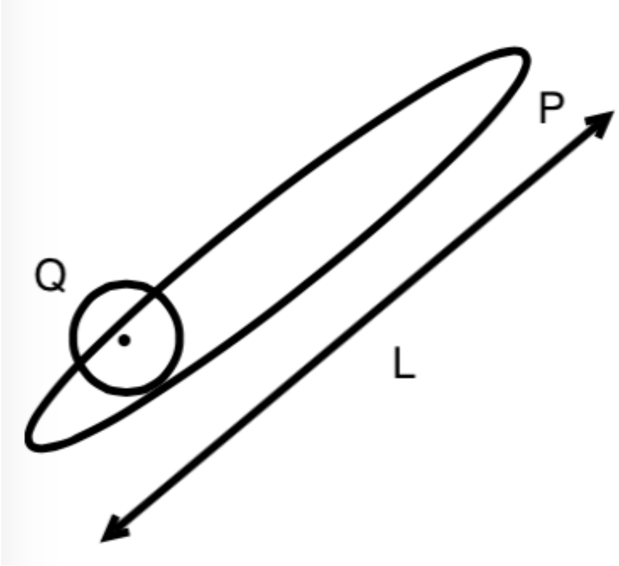
Proposal distribution has significant effect on performance

Common proposal is Gaussian centered on current state

$$Q(x'|x) = N(x,\sigma^2)$$

- Small σ : slow random walk with long correlation times

  $\sim (L/\sigma)^2$ iterations
- Large σ : many rejections

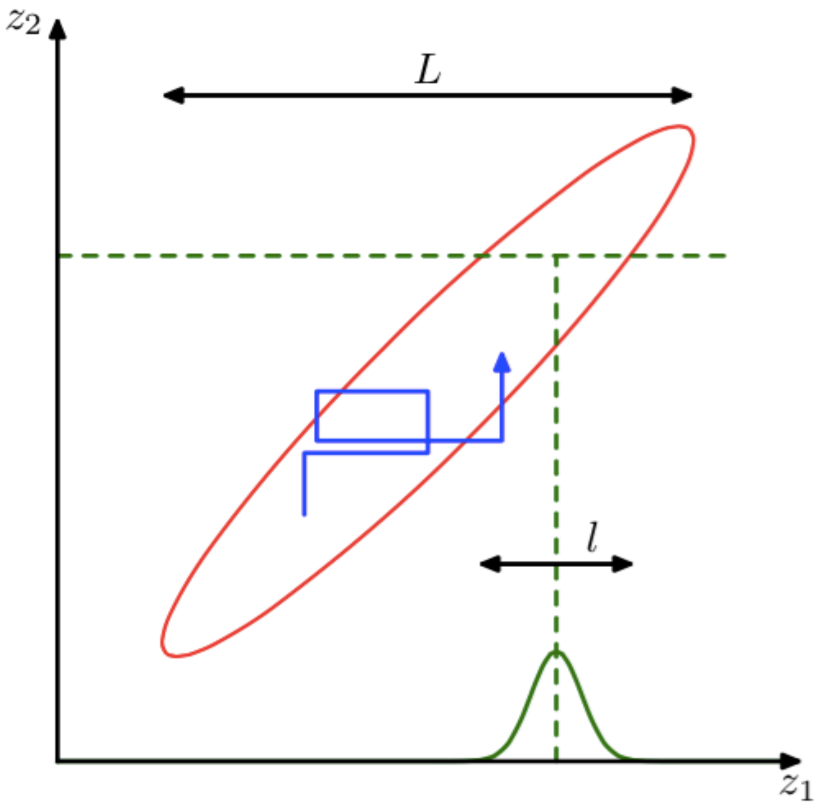


Bottom line: if length scales over which the distributions vary are very different across dimensions, then M-H may converge slowly

Gibbs sampling

A simple, general MCMC algorithm, with no rejections

- Initialize **x** to some value
- Pick each variable in turn, or randomly, and resample $P(x_i|\mathbf{x}_{-i})$



Step size governed by conditional distribution – need O$((L/l)^2)$ steps to get independent samples from distribution

Gibbs sampling is easy

Based on convenient features of conditional distributions

Conditionals w/ a few discrete settings can be explicitly normalized

$$P(x_i|\mathbf{x}_{-i}) \propto P(x_i, \mathbf{x}_{-i})$$

$$= \frac{P(x_i, \mathbf{x}_{-i})}{\sum_{x_i'} P(x_i', \mathbf{x}_{-i})}$$

Continuous conditionals only univariate

→ amenable to standard sampling methods

Proof that Gibbs sampling is valid

Check detailed balance for component update

Metropolis-Hastings proposals: $Q(\mathbf{x}'|\mathbf{x}) = P(x'_i|\mathbf{x}_{-i})$

acceptance probability:

$$\frac{P(\mathbf{x}')Q(\mathbf{x}|\mathbf{x}')}{P(\mathbf{x})Q(\mathbf{x}'|\mathbf{x})}$$

$$= \frac{P(x'_i|\mathbf{x}'_{-i})P(\mathbf{x}'_{-i})P(x_i|\mathbf{x}'_{-1})}{P(x_i|\mathbf{x}_{-i})P(\mathbf{x}_{-i})P(x'_i|\mathbf{x}_{-i})} = 1$$

→ accept with probability 1

Can apply a series of these operators, without needing to check acceptance

Summary

We need approximate methods to solve sums/integrals

Monte Carlo does not explicitly depend on dimension, although simple methods work only in low dimensions

Markov chain Monte Carlo (MCMC) can make local moves. By assuming less it is more applicable to higher dimensions

It produces approximate, correlated samples

Simple computations → easy to implement

Comparison to variational inference

Benefits of variational inference:

- Fast for smallish problems
- Deterministic
- Easy to know when to stop
- Provides lower bound on log likelihood

Benefits of sampling:

- Easier to implement
- Applicable to broader class of models (without nice conjugate priors)
- Can be faster when applied to large models/datasets