Stat 521A
Lecture 11

Outline

- Forward sampling (12.1)
- Importance sampling (12.2)
- MCMC (12.3)
- Collapsed particles (12.4)
- Deterministic search (12.5)

Monte Carlo integration

- The goal is to approximate E[f(X)] for some function f eg f(X) = I(X_i=k), so E[f(X)] = p(X_i=k)
- Usually we take expectations wrt p(X|e), where e is the evidence
- If we can draw samples X ~ p(X|e), we can evaluate the expectation thus:

$$E_P[f] \approx \frac{1}{M} \sum_{m=1}^{M} f(x[m]).$$

Error analysis

Let $\mu = E[h(X)]$ be the exact expected value, and $\hat{f}_S$ a Monte Carlo approximation based on S samples. One can show a central-limit type theorem

$$(\hat{f}_S - \mu) \rightarrow \mathcal{N}(0, \frac{\sigma^2}{S}) \tag{16.6}$$

where $\sigma^2 = \text{Var}\,[h(X)]$. The latter quantity can itself be estimated by MC:

$$\hat{\sigma}^2 = \frac{1}{S}\sum_{s=1}^{S}(f(\theta^s) - \hat{f}_S)^2 \tag{16.7}$$

Then we have

$$P\left\{\mu - 1.96\frac{\hat{\sigma}^2}{\sqrt{S}} \leq \hat{f}_s \leq \mu + 1.96\frac{\hat{\sigma}^2}{\sqrt{S}}\right\} \approx 0.95 \tag{16.8}$$

The term $\sqrt{\frac{\hat{\sigma}^2}{S}}$ is called the (numerical or empirical) **standard error**. Thus we see that the error in our MC estimate goes down at a rate of $1/\sqrt{S}$.

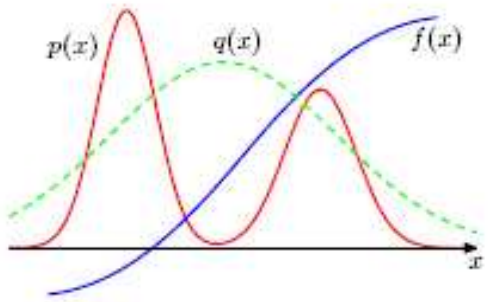
Forward sampling

- To sample from the prior p(x) of a DGM is easy: just sample each node in topological order, conditional on its parents
- To sample from the prior of a UGM is much harder
- Usually we want to sample from the posterior p(x|e)
- We can use forwards sampling and throw away all samples that are inconsistent with e; this is called **rejection sampling** ("logic sampling" in the context of discrete DGMs) and is very inefficient

Unnormalized importance sampling

- Often sampling from P is hard
- Suppose we sample from a proposal distribution Q instead. All we require is that P(x)>0 => Q(x)>0

$$
\begin{aligned}
\boldsymbol{E}_{Q(\boldsymbol{X})}\left[f(\boldsymbol{X})\frac{P(\boldsymbol{X})}{Q(\boldsymbol{X})}\right] &= \sum_{\boldsymbol{x}} Q(\boldsymbol{x}) f(\boldsymbol{x}) \frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})} \\
&= \sum_{\boldsymbol{x}} f(\boldsymbol{x}) P(\boldsymbol{x}) \\
&= \boldsymbol{E}_{P(\boldsymbol{X})}[f(\boldsymbol{X})]
\end{aligned}
$$



$$
\hat{\boldsymbol{E}}_{\mathcal{D}}(f) = \frac{1}{M}\sum_{m=1}^{M} f(\boldsymbol{x}[m])\frac{P(\boldsymbol{x}[m])}{Q(\boldsymbol{x}[m])}.
$$

Unbiased estimator

Variance

- Variance of estimator given by

$$\begin{aligned}\sigma_Q^2 &= E_{Q(\boldsymbol{X})}\left[(f(\boldsymbol{X})w(\boldsymbol{X}))^2\right] - E_{Q(\boldsymbol{X})}\left[(f(\boldsymbol{X})w(\boldsymbol{X}))\right]^2 \\ &= E_{Q(\boldsymbol{X})}\left[(f(\boldsymbol{X})w(\boldsymbol{X}))^2\right] - (E_{P(\boldsymbol{X})}[f(\boldsymbol{X})])^2.\end{aligned}$$

- Let f(X)=1. Then variance is variance of P(X)/Q(X)

$$E_{Q(\boldsymbol{X})}\left[\left(\frac{P(\boldsymbol{X})}{Q(\boldsymbol{X})}\right)^2\right] - \left(E_{Q(\boldsymbol{X})}\left[\frac{P(\boldsymbol{X})}{Q(\boldsymbol{X})}\right]\right)^2,$$

- Variance will be large if Q(x) << P(x) f(x)

Normalized importance sampling

- Often we only know P'(x) = α P(x) with unknown α
- Define

$$w(\boldsymbol{X}) = \frac{\tilde{P}(\boldsymbol{X})}{Q(\boldsymbol{X})}.$$

- Then

$$E_{Q(\boldsymbol{X})}[w(\boldsymbol{X})] = \sum_{\boldsymbol{x}} Q(\boldsymbol{x}) \frac{\tilde{P}(\boldsymbol{x})}{Q(\boldsymbol{x})} = \sum_{\boldsymbol{x}} \tilde{P}(\boldsymbol{x}) = \alpha.$$

$$\begin{aligned}
E_{P(\boldsymbol{X})}[f(\boldsymbol{X})] &= \sum_{\boldsymbol{x}} P(\boldsymbol{x}) f(\boldsymbol{x}) \\
&= \sum_{\boldsymbol{x}} Q(\boldsymbol{x}) f(\boldsymbol{x}) \frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})} \\
&= \frac{1}{\alpha} \sum_{\boldsymbol{x}} Q(\boldsymbol{x}) f(\boldsymbol{x}) \frac{\tilde{P}(\boldsymbol{x})}{Q(\boldsymbol{x})} \\
&= \frac{1}{\alpha} E_{Q(\boldsymbol{X})}[f(\boldsymbol{X}) w(\boldsymbol{X})] \\
&= \frac{E_{Q(\boldsymbol{X})}[f(\boldsymbol{X}) w(\boldsymbol{X})]}{E_{Q(\boldsymbol{X})}[w(\boldsymbol{X})]}
\end{aligned}$$

$$\hat{E}_{\mathcal{D}}(f) = \frac{\sum_{m=1}^{M} f(\boldsymbol{x}[m]) w(\boldsymbol{x}[m])}{\sum_{m=1}^{M} w(\boldsymbol{x}[m])}$$

Bias

- Biased estimator

$$\hat{E}_{\mathcal{D}}(f) = \frac{\sum_{m=1}^{M} f(\boldsymbol{x}[m])w(\boldsymbol{x}[m])}{\sum_{m=1}^{M} w(\boldsymbol{x}[m])}$$

- Eg M=1. x[1] ~ Q has wrong mean

$$\frac{f(\boldsymbol{x}[1])w(\boldsymbol{x}[1])}{w(\boldsymbol{x}[1])} = f(\boldsymbol{x}[1]).$$

- But bias -> 0 as 1/M since numerator and denominator are both unbiased

Variance

- Variance ->0 as 1/M

$$\mathbb{V}\mathrm{ar}_P\left[\hat{E}_{\mathcal{D}}(f(\boldsymbol{X}))\right] \approx \frac{1}{M}\mathbb{V}\mathrm{ar}_P[f(\boldsymbol{X})](1 + \mathbb{V}\mathrm{ar}_Q[w(\boldsymbol{X})]),$$

- Variance of optimal estimator is $\hat{\mathbb{V}}\mathrm{ar}_P[f(\boldsymbol{X})]/M$
- Ratio is $\dfrac{1}{1 + \mathbb{V}\mathrm{ar}_Q[w(\boldsymbol{x})]}$.

- Effective sample size

$$\begin{aligned} M_{\text{eff}} &= \frac{M}{1 + \mathbb{V}\mathrm{ar}[\mathcal{D}]} \\ \mathbb{V}\mathrm{ar}[\mathcal{D}] &= \sum_{m=1}^{M} w(\boldsymbol{x}[m])^2 - (\sum_{m=1}^{M} w(\boldsymbol{x}[m]))^2. \end{aligned}$$

Likelihood weighting

- Let us apply importance sampling to a DGM where the proposal is as follows: do forwards in the mutilated DGM where observed nodes are clamped to Z=z
- Prop 12.2.5. Weights are $w(\xi) = \frac{P_{\mathcal{B}}(\xi)}{P_{\mathcal{B}_{\boldsymbol{Z}=\boldsymbol{z}}}(\xi)}$.

Algorithm 12.2 Likelihood Weighted Particle Generation

```
Procedure LW-sample (
  B,     // Bayesian network over X
  Z = z  // Event in the network
)
1   Let X_1, ..., X_n be a topological ordering of X
2   w ← 1
3   for i = 1, ..., n
4     u_i ← x⟨Pa_{X_i}⟩   // Assignment to Pa_{X_i} in x_1, ..., x_{i-1}
5     if X_i ∉ Z then
6       Sample x_i from P(X_i | u_i)
7     else
8       x_i ← z⟨X_i⟩   // Assignment to X_i in z
9       w ← w · P(x_i | u_i)   // Multiply weight by probability of desired value
10  return (x_1, ..., x_n), w
```

Using LW weights

- Recall that E[w(X)] = α = p(Z=z)
- Ratio likelihood weighting: run LW twice for each y

$$\hat{P}_{\mathcal{D}}(y \mid e) = \frac{\hat{P}_{\mathcal{D}}(y, e)}{\hat{P}_{\mathcal{D}'}(e)} = \frac{1/M \sum_{m=1}^{M} w[m]}{1/M' \sum_{m=1}^{M'} w'[m]}.$$

- Normalized likelihood weighting: run LW once, and use samples to evaluate any query

$$\hat{P}_{\mathcal{D}}(y \mid e) = \frac{\sum_{m=1}^{M} w[m] \mathbf{1}\{y[m] = y\}}{\sum_{m=1}^{M} w[m]}.$$

= p(y,z)/ p(z)

Efficiency

- Although LW does not “throw away” samples that are inconsistent with e, it down weights them
- If the evidence is at the leaves, the samples are drawn from the prior and may be assigned low weight
- Backward importance sampling (evidence reversal): if X->Y=y, sample from $Q(X) \propto p(Y=y|X)$
- Importance sampling does not scale well to high dimensions, because hard to make Q match P

MCMC

- Markov Chain Monte Carlo constructs a Markov chain whose stationary distribution is equal to the posterior p(x|e).
- Metropolis Hastings: only need proposal Q(x'|x) and ability to evaluate $\pi(x) = p(x,e) \propto p(x|e)$
- Gibbs: only need ability to sample full conditionals p(xi|x(-i),e)

Metropolis Hastings algorithm

- We propose q(x'|x) and evaluate $\alpha=\pi(x')/\pi(x)$
- If α >= 1, we accept, otherwise we accept wp r
- Always accept uphill move, occasionally accept downhill move
- If proposal is asymmetric, need Hastings correction

$$\begin{aligned} \alpha &= \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)} = \frac{\pi(x')/q(x'|x)}{\pi(x)/q(x|x')} \\ r &= \min(1,\alpha) \end{aligned}$$

MH pseudocode

1 Initialize x^0
2 **for** $s = 0, 1, 2, \ldots$ **do**
3 Sample $x' \sim q(x'|x)$
4 Compute acceptance probability

$$\alpha = \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)} = \frac{\pi(x')/q(x'|x)}{\pi(x)/q(x|x')}$$

5 Compute $r = \min(1, \alpha)$
6 Set new sample to

$$x^{s+1} = \begin{cases} x' & \text{with probability } r \\ x^s & \text{with probability } 1 - r \end{cases}$$

Why MH works

- MH generates a MC with this transition matrix

$$p(x'|x) = \begin{cases} q(x'|x)r(x'|x) & \text{if } x' \neq x \\ q(x|x) + \sum_{x' \neq x} q(x'|x)(1 - r(x'|x)) & \text{otherwise} \end{cases} \quad (16.21)$$

Theorem 16.2.1. *If the transition matrix defined by the MH algorithm (given by Equation 16.21) is ergodic and irreducible, then π is its unique limiting distribution.*

Proof. Consider two states x and x'. Either

$$\pi(x)q(x'|x) < \pi(x')q(x|x') \quad (16.22)$$

or

$$\pi(x)q(x'|x) > \pi(x')q(x|x') \quad (16.23)$$

We will ignore ties (which occur with probability zero for continuous distributions). Without loss of generality, assume that $\pi(x)q(x'|x) > \pi(x')q(x|x')$. Hence

$$\alpha(x'|x) = \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)} < 1 \quad (16.24)$$

Proof cont'd

Hence we have $r(x'|x) = \alpha(x'|x)$ and $r(x|x') = 1$.

Now to move from x to x' we must first propose x' and then accept it. Hence

$$p(x'|x) = q(x'|x)r(x'|x) = q(x'|x)\frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)} = \frac{\pi(x')}{\pi(x)}q(x|x') \tag{16.25}$$

Hence

$$\pi(x)p(x'|x) = \pi(x')q(x|x') \tag{16.26}$$

The backwards probability is

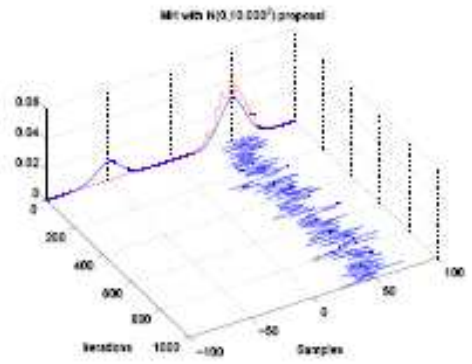
$$p(x|x') = q(x|x')r(x|x') = q(x|x') \tag{16.27}$$

since $r(x|x') = 1$. Inserting this into Equation 16.26 we get

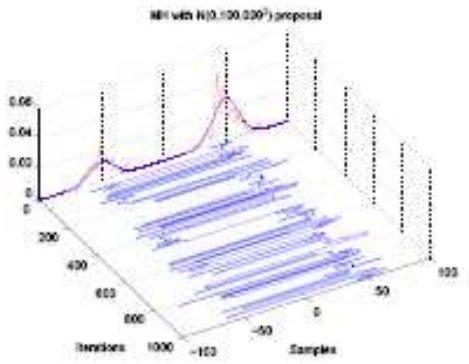
$$\pi(x)p(x'|x) = \pi(x')p(x|x') \tag{16.28}$$

so detailed balance holds. Hence, from Theorem ??, π is the stationary distribution. ∎

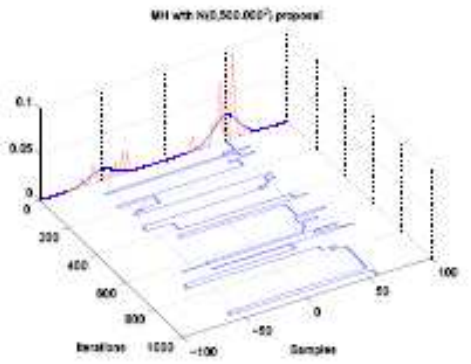
Proposal distributions



(a)

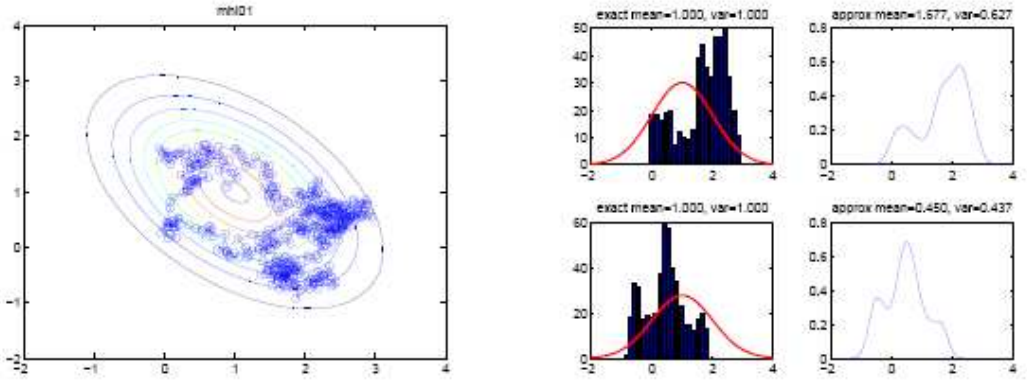


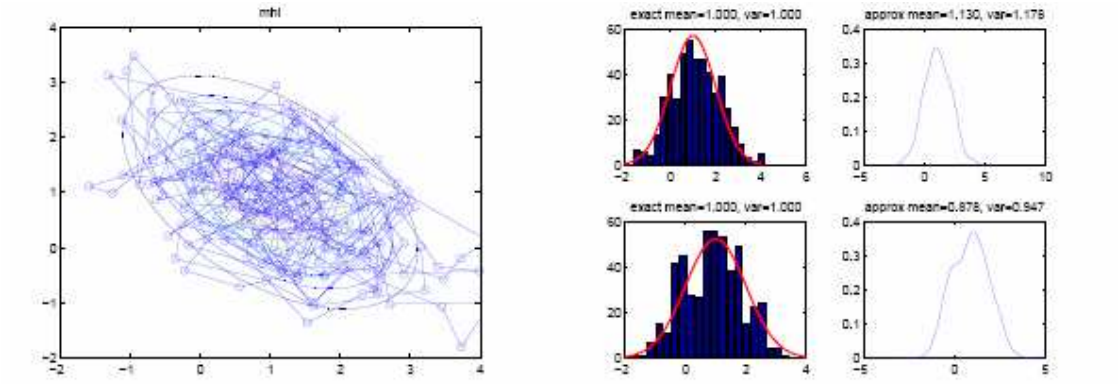
(b)



(c)

Proposal distributions





Methods for choosing proposals

- Initialize chain at a local mode (found with an optimizer)
- Gaussian random walk, with covariance = Hessian
- Mixture of base kernels, corresponding to different heuristic algorithms

$$q(x'|x) = \sum_{k=1}^{K} w_k q_k(x'|x)$$

- Adaptive MCMC: modify Gaussian covariance online

Gibbs sampling

- Sample each node given all others, from its full conditional

1. $x_1^{s+1} \sim p(x_1|x_2^s, \ldots, x_d^s)$
2. $x_2^{s+1} \sim p(x_2|x_1^{s+1}, x_3^s, \ldots, x_d^s)$
3. $x_i^{s+1} \sim p(x_i|x_{1:i-1}^{s+1}, x_{i+1:d}^s)$
4. $x_d^{s+1} \sim p(x_d|x_1^{s+1}, \ldots, x_{d-1}^{s+1})$

- This is MH with the following proposal

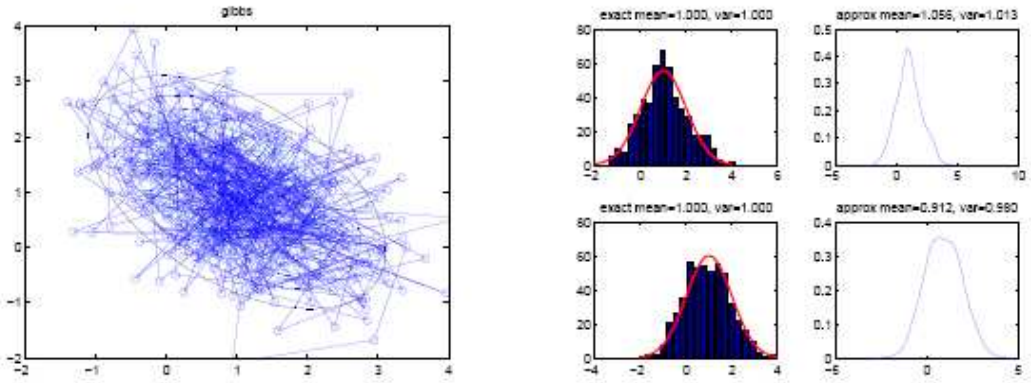
$$q((x_i', \mathbf{x}_{-i})|(x_i, \mathbf{x}_{-i})) = p(x_i'|\mathbf{x}_{-i})$$

- Acceptance rate is 100%

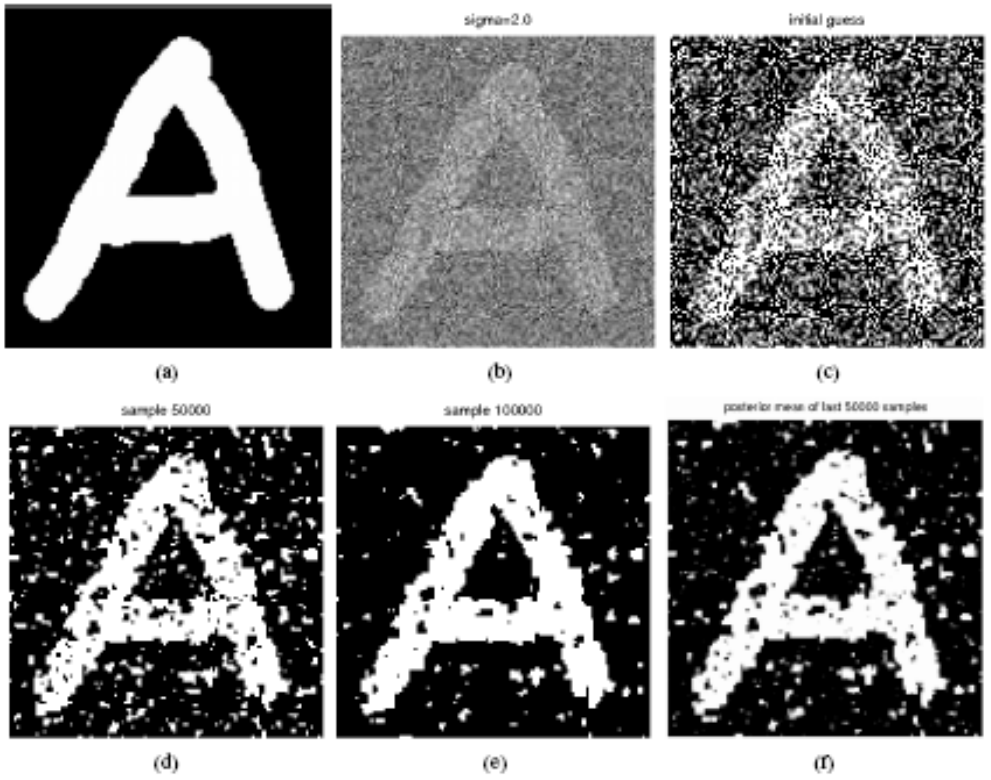
$$\alpha = \frac{p(\mathbf{x}')q(\mathbf{x}|\mathbf{x}')}{p(\mathbf{x})q(\mathbf{x}'|\mathbf{x})} = \frac{p(x_i'|\mathbf{x}_{-i})p(\mathbf{x}_{-i})p(x_i|\mathbf{x}_{-i})}{p(x_i|\mathbf{x}_{-i})p(\mathbf{x}_{-i})p(x_i'|\mathbf{x}_{-i})} = 1$$

Gibbs for bivariate Gaussian

$$
\begin{aligned}
p(x_1|x_2) &= \mathcal{N}(x_1|\mu_{1|2}, \Sigma_{1|2}) \\
\mu_{1|2} &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) \\
\Sigma_{1|2} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}
\end{aligned}
$$



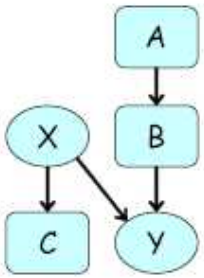
Gibbs for Ising



$$p(x_t = +1 | \mathrm{x}_{-t}, \mathrm{y}, \theta) = \frac{\exp[J w_t] \phi_t(+1, y_t)}{\exp[J w_t] \phi_t(+1, y_t) + \exp[-J w_t] \phi_t(-1, y_t)}$$
$$= \sigma(2J \log \frac{\phi_t(+1)}{\phi_t(-1)})$$

BUGS

- Bayesian Updating using Gibbs Sampling

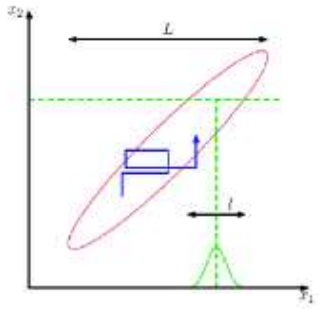



```
var
   A, B, C, X, Y, mu, tau, p[2,3], q;

p = ...
A ~ dbern(0.3)
B ~ dcat(p[A,1:3])
X ~ dnorm(-1,0.25)
mu <- 3*X+B^2
tau <- 1/X^2
Y ~ dnorm(mu,tau)
logit(q) <- 4*X + 2
C ~ dbern(q)
```

Single vs block updates

- Gibbs does single site updating which can move slowly, or even get stuck (eg XOR)
- Blocked Gibbs sampling samples multiple variables at once



Accuracy

- Even though the samples are correlated, we have a CLT-type result

$$(\mu - \hat{\mu}) \to \mathcal{N}(0, \sigma^2)$$

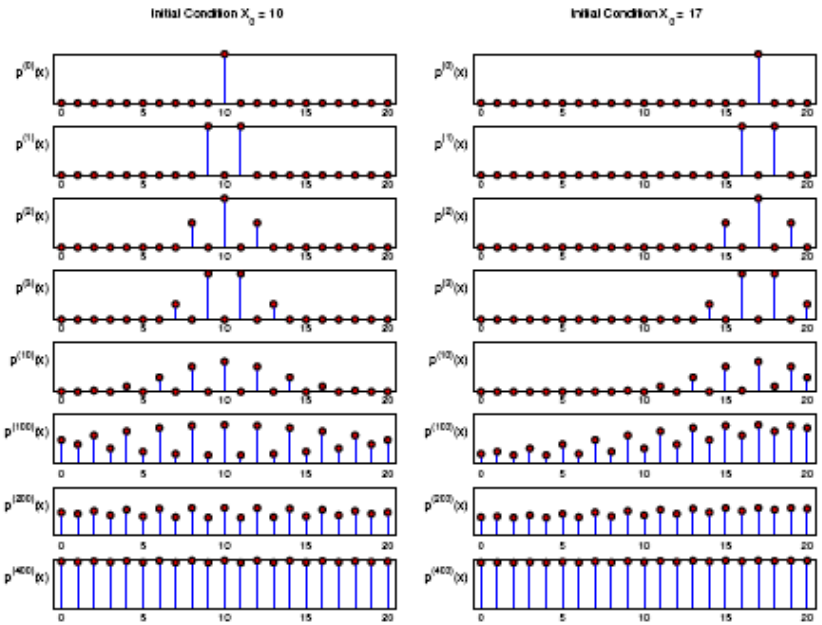
$$\sigma^2 = \text{Var}\left[f(X)\right] + 2\sum_{\ell=0}^{\infty} \text{Cov}[f(X_t), f(X_{t+\ell})]$$

- Autocorrelation function

$$\rho(\ell) = \frac{\text{Cov}[f(X_t), f(X_{t+\ell})]}{\sigma^2}$$

Mixing time

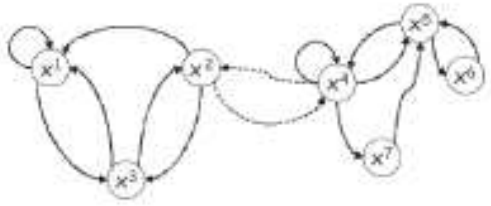
- Mixing time is time to reach stationary distribution



Samples drawn before convergence (during burnin phase) should be discarded

Conductance

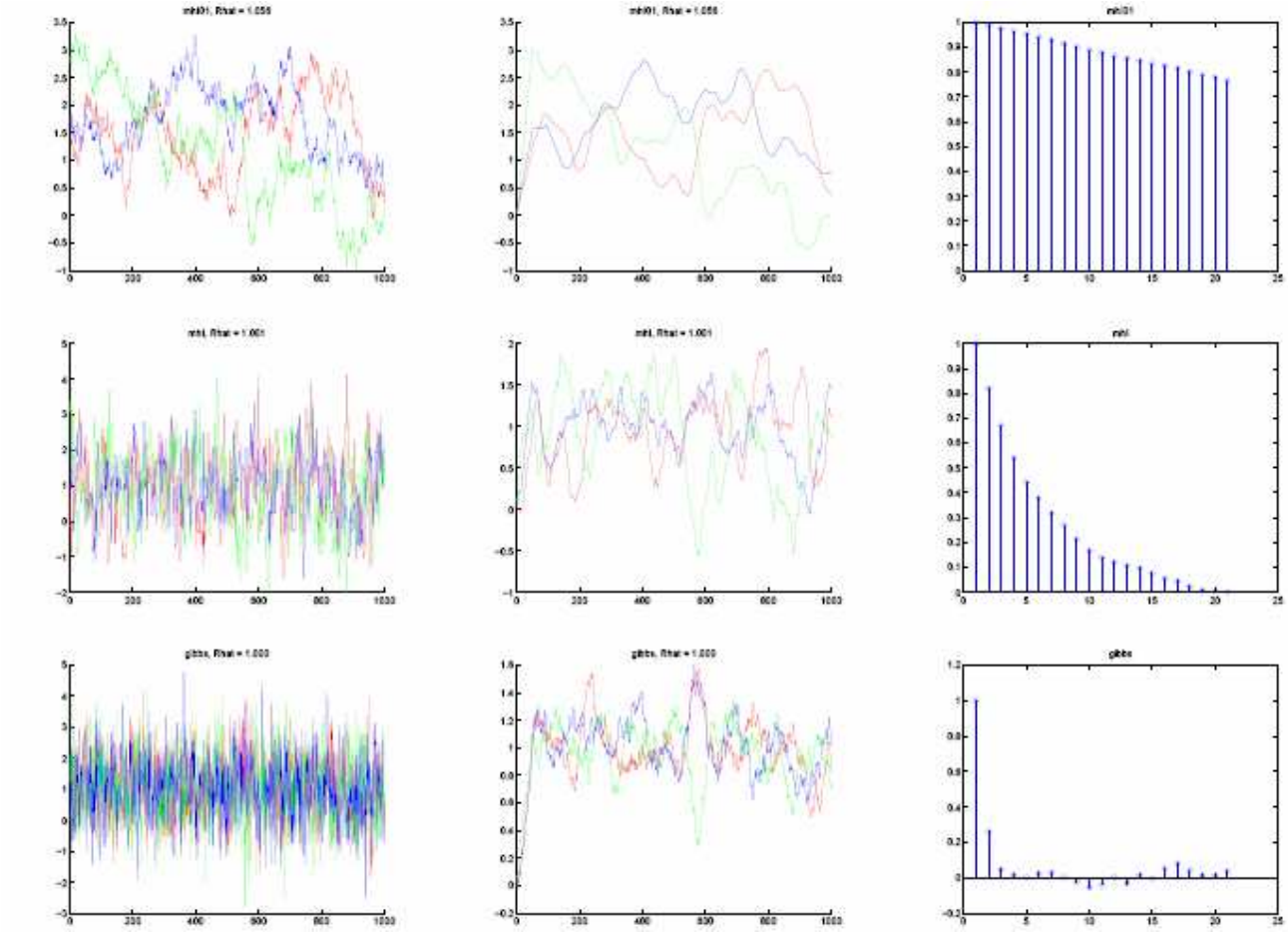
- Mixing time depends on eigengap, $\gamma=\lambda_1-\lambda_2$
- Hard to compute
- Can develop bounds based on the conductance (which is low if there are narrow bottlenecks in the state space)



Convergence

- 2 issues
  - Speeding up convergence
  - Determining if convergence has happened
- Speedups: various tricks, see later
- Determining: various heuristics

Traceplots and ACF



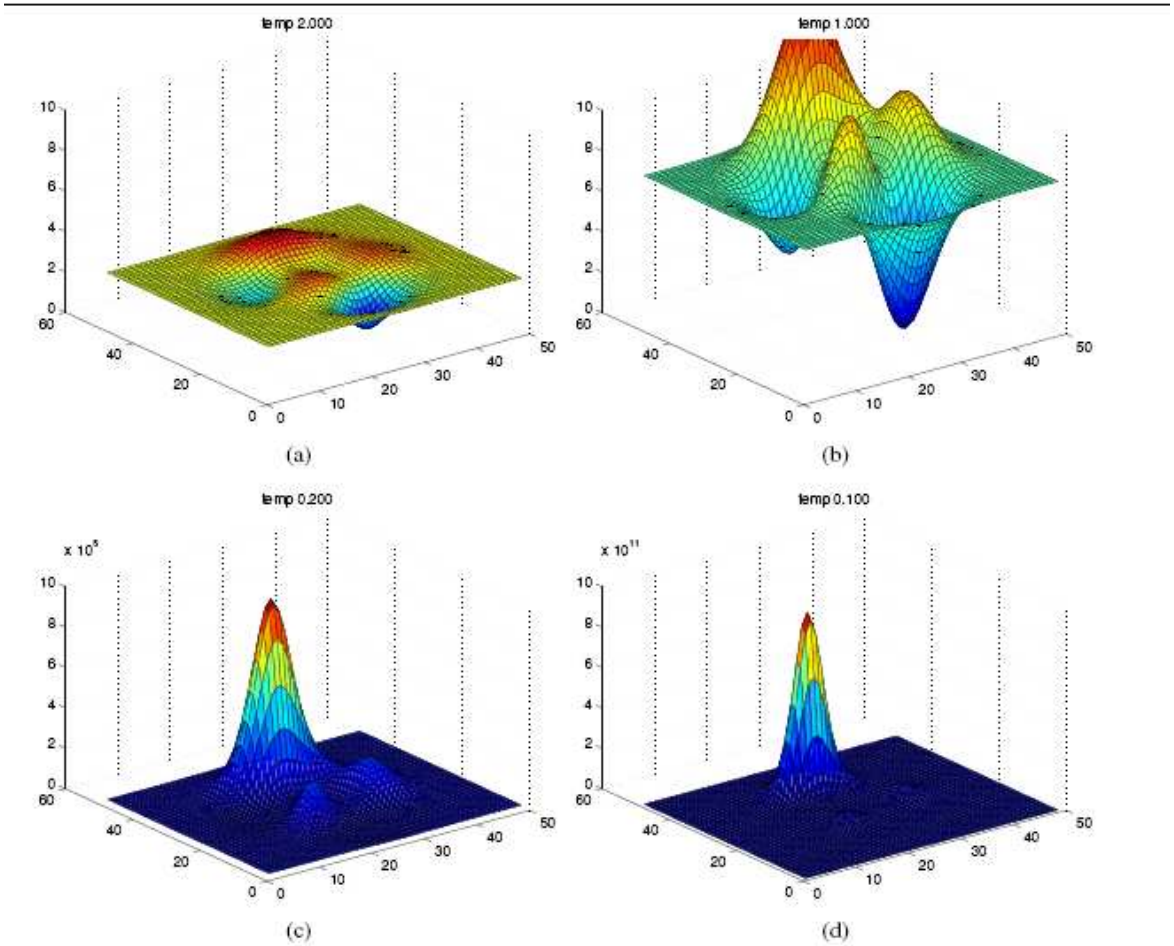
EPSR

- Start 3 chains from different states, run them for a while, check if variance within a chain is comparable to variance between chains
- Can be formalized using the Rhat statistic (estimated potential scale reduction).
- If Rhat ~ 1.0 for a specific f(X), then it suggest that the chain has converged.
- Can compute Rhat for multiple features f(X).

Simulated annealing

- Global optimization method
- Raise surface to a temperature to smooth it out/ kill off the non-peaks

$$\pi_s(x) = \pi(x)^{1/T_s}$$



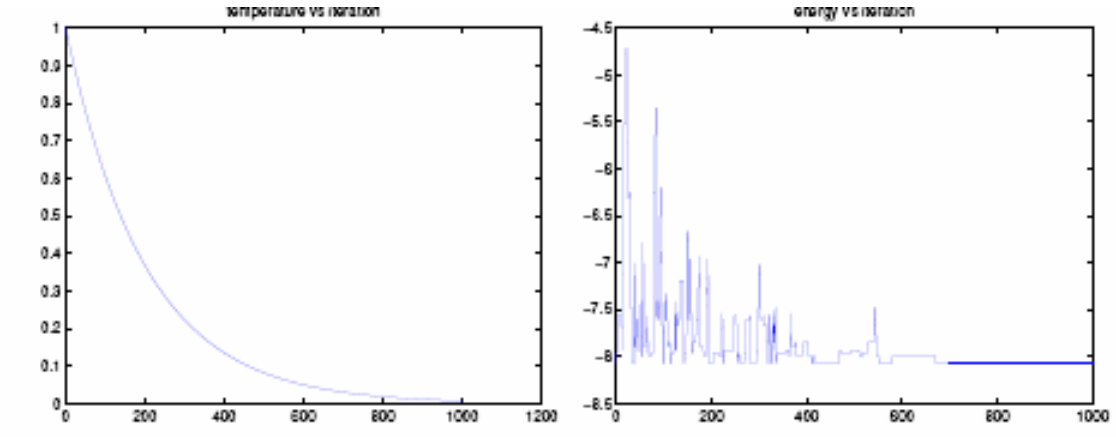
Simulated annealing

- $\pi(x) = \exp(-E(x))$, E(x)=energy (+ve or –ve)

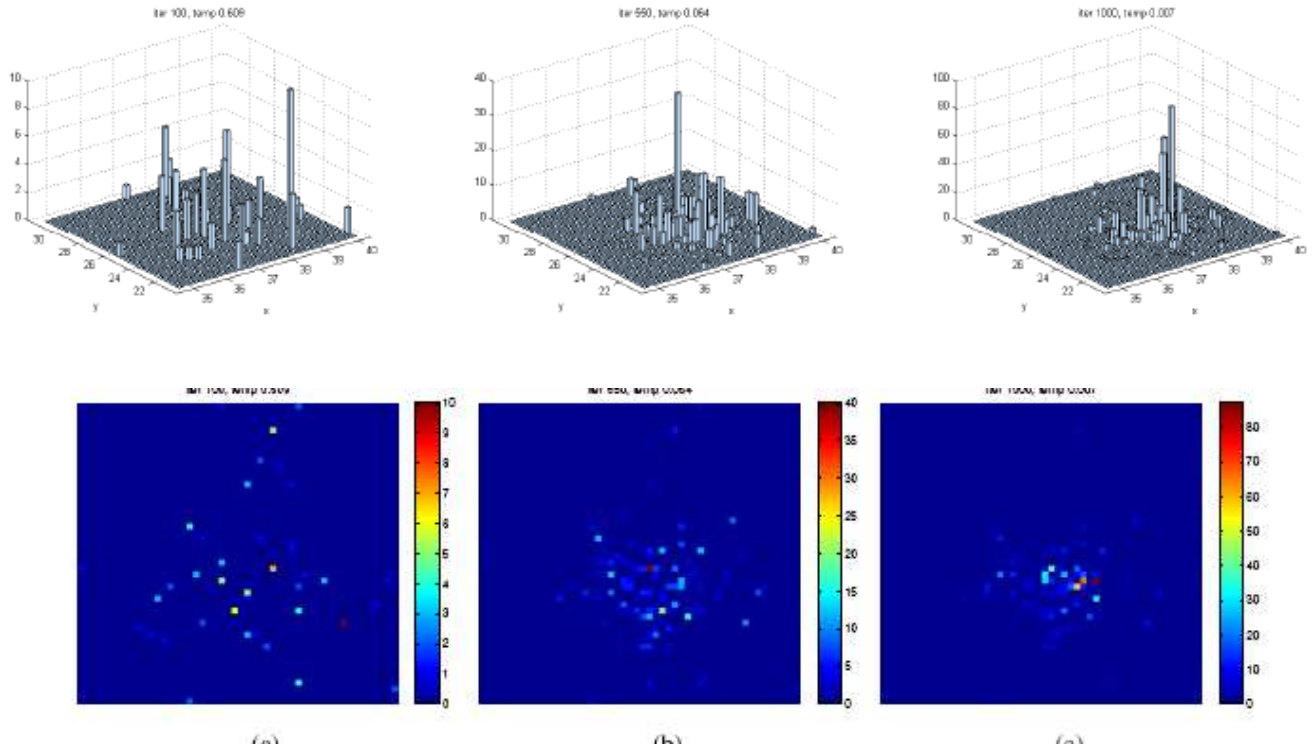
$$\begin{aligned}\alpha &= \frac{\pi(x')^{1/T_s}}{\pi(x)^{1/T_s}} \\ &= \frac{\exp(-E(x'))^{1/T_s}}{\exp(-E(x))^{1/T_s}} \\ &= \exp\left((E(x) - E(x'))/T_s\right)\end{aligned}$$

- Cooling schedule

$$T_s = T_0 C^s$$



Samples from SA



Parallel tempering

- Run multiple chains at different temperatures
- Let them swap samples
- Lowest chain at temp=1 is used to return samples to user; other chains encourage global moves
- Good for multi-model posteriors

Evolutionary Monte Carlo

- Combine ideas from genetic algorithms with MCMC
- Population is the new state space; propose moves that swap pieces of particles.

GMs and MCMC

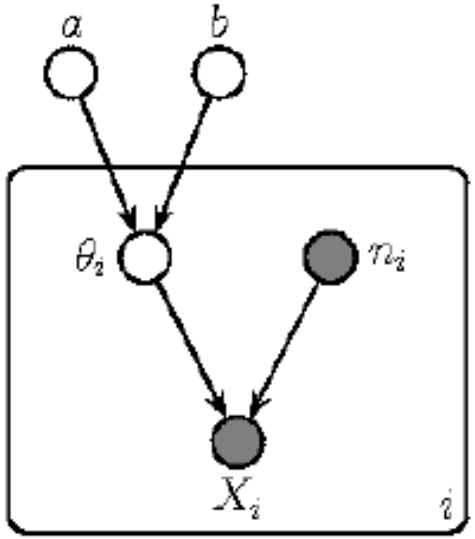
- MCMC can benefit from GMs
  - To define Markov blanket for Gibbs
  - To efficiently evaluate $\pi(x')/\pi(x)$ for MH
- GMs need MCMC for
  - State estimation (Inference)
  - Parameter estimation (Learnign)
  - Model selection (structure learning)

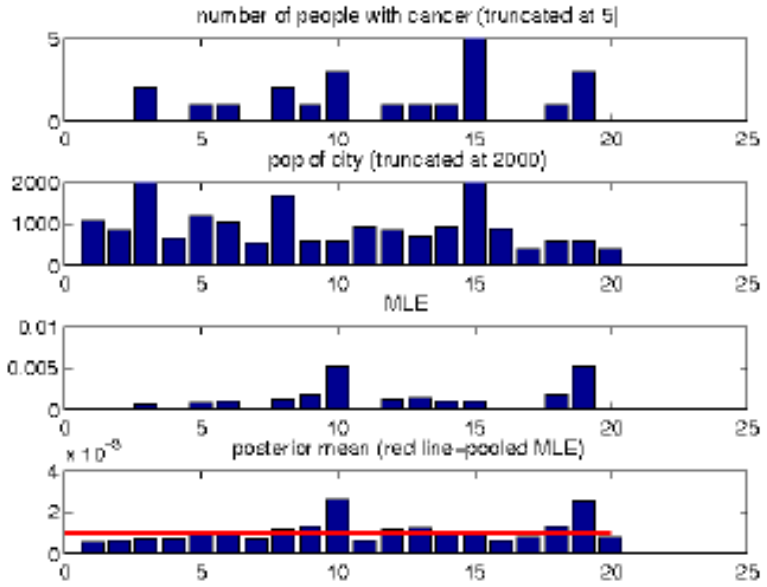
Collapsed samplers

- A collapsed sampler means analytically integrating out some variables and sampling the rest
- Aka Rao-Blackwellization
- Later we will see an interesting example when we consider RB for particle filtering
- Today, a simpler example, which will form the basis of a homework exercise

Hierarchical Bayesian modeling

- Model related cancer incidence rates



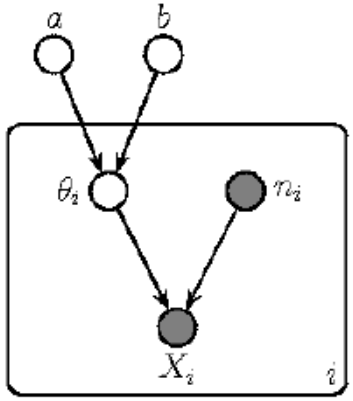


$$p(\mathbf{x}, \mathbf{n}, \boldsymbol{\theta}, a, b) = \prod_{i=1}^{n} p(x_i|n_i, \theta_i)p(\theta_i|a, b)p(a, b) \quad (1)$$

$$= \prod_{i=1}^{n} \text{Bin}(x_i|n_i, \theta_i)\text{Beta}(\theta_i|a, b)p(a, b) \quad (2)$$

Inference

- Gibbs sampling $p(a,b,\theta_i|D)$ - homework
- MH $p(a,b|D)$ – sample a,b, integrate out theta



$$p(\boldsymbol{\alpha}|\mathcal{D}) \propto p(\boldsymbol{\alpha}) \prod_i \int p(x_i|n_i,\theta_i) p(\theta_i|a,b) d\theta_i$$
$$= p(\boldsymbol{\alpha}) \prod_i \frac{B(a+x_i, b+n_i-x_i)}{B(a,b)}$$

$$E[\theta_i|\mathcal{D}] = E\left[E[\theta_i|\boldsymbol{\alpha},\mathcal{D}]|\mathcal{D}\right] \approx \frac{1}{S}\sum_{s=1}^{S} E[\theta_i|\boldsymbol{\alpha}^s]$$

- Empirical Bayes (a^*,b^*)=arg max $p(a,b|D)$, then $E[\text{theta}_i|a^*,b^*]$

MH for Missouri cancer problem

- We use mean m=a/(a+b) and K=a+b
- Beta prior on m, noninformative prior on K

$$p(m, K|\mathcal{D}) \propto \frac{m^{a_m-1}(1-m)^{b_m-1}}{(1+K)^2} \prod_i \frac{B(Km+x_i, K(1-m)+n_i-y_i)}{B(Km, K(1-m))}$$

- Transform to unconstrained params

$$\theta_1 = \log \frac{m}{1-m}, \quad \theta_2 = \log K$$

- MH with diagonal Gaussian proposal

Inference in discrete state spaces

- For a cts state space, $\pi(x)$ is a pdf, so we represent high probability values by repeating them many times
- For a discrete state space (eg model search, or after integrating out cts), the posterior is a pmf, so we can evaluate p(x|e) up to a normalization constant. There is no need to repeat a discrete state to represent its probability.
- Hence it is better to rapidly visit as many states as possible, and *never revisit a state*
- Hence use stochastic/ deterministic, local/ global search not MCMC

Deterministic search

- There are many (exact or approx) methods from the AI/ OR communities to find the top K values of a discrete distribution
- We approximate P(Z=z) by counting how many instantiations are compatible with Z=z, weighted by their probability

$$\sum_{m=1}^{M} \mathbf{1}\{x[m] = z\}\tilde{P}(\xi[m]),$$

- More precisely, we have bounds on p(Z=z)

$$\sum_{m=1}^{M} \mathbf{1}\{x[m] = z\}\tilde{P}(\xi[m]) \leq \tilde{P}(Z = z) \leq \left(1 - \sum_{m=1}^{M} \mathbf{1}\{x[m] \neq z\}\tilde{P}(\xi[m])\right).$$

Bounds on conditional probabilities

- We have

$$\begin{array}{ccccc} \ell_{\boldsymbol{y},\boldsymbol{e}} & \leq & P(y,e) & \leq & u_{\boldsymbol{y},\boldsymbol{e}} \\ \ell_{\boldsymbol{e}} & \leq & P(e) & \leq & u_{\boldsymbol{e}} \end{array}$$

$$\frac{\ell_{\boldsymbol{y},\boldsymbol{e}}}{u_{\boldsymbol{e}}} \leq P(y \mid e) \leq \frac{u_{\boldsymbol{y},\boldsymbol{e}}}{\ell_{\boldsymbol{e}}}.$$