DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF COPENHAGEN

Bayesian Inference and Sampling From Probability Distributions

Kim Steenstrup Pedersen

Plan for this lecture

- Problems of Bayesian inference
- MAP point estimate: Gradient-based optimization
- Basic sampling methods
  - Rejection sampling
- Markov Chain Monte Carlo (MCMC) methods
  - Metropolis-Hastings algorithm

Problems of Bayesian Inference

(Rogers & Girolami, Ch. 4.1-4.2)

Problems of Bayesian inference

- The Bayesian formulation for both regression and classification (see binary classifier example in the book) involves the posterior distribution over model parameters

Likelihood

Prior

Posterior

$$p(\mathbf{w}|\mathbf{t},\mathbf{X},\theta) = \frac{p(\mathbf{t}|\mathbf{X},\mathbf{w})p(\mathbf{w}|\theta)}{p(\mathbf{t}|\mathbf{X})}$$

Prior parameters / hyperparameters

$$\mathbf{t} = (t_1,\ldots,t_N)^T, \mathbf{X} = (\mathbf{x}_1,\ldots,\mathbf{x}_N)^T, \mathbf{w} \in R^D, \theta \in R^M$$

- Only in specific cases, such as when we use conjugate likelihood-prior pairs, can we evaluate the posterior exactly. The problem is the integral in the marginal likelihood

$$p(\mathbf{t}\,|\,\mathbf{X}) = \int p(\mathbf{t}\,|\,\mathbf{X},\mathbf{w})\,p(\mathbf{w}\,|\,\theta)\,d\mathbf{w}$$

Problems of Bayesian inference

- Usually we would like to be able to use our Bayesian model to make prediction (inference) on new unseen data (t_{new}, x_{new})
- In Bayesian inference, we ideally want to use the complete posterior distribution over parameters (instead of using a single parameter point estimate). We want to compute the predictive distribution

$$p(t_{new} \mid x_{new}, \mathbf{t}, \mathbf{X}, \theta) = E_{p(\mathbf{w}|\mathbf{t},\mathbf{X},\theta)}\left[p(t_{new} \mid x_{new}, \mathbf{w}) \right]$$

Expectation with respect to the posterior

Likelihood for the new data

Possible solutions

- Find a local maximum of the posterior and use as point estimate of the parameters (MAP point estimate)
- Local approximation of the posterior distribution (we skip this part on this course)
- Sampling from the posterior distribution

Sampling for Bayesian inference

- For general Bayesian models, we can approximate the expectation in the predictive distribution by sampling N_s samples of parameter vectors $\mathbf{w}_s$ from the posterior and use this approximation

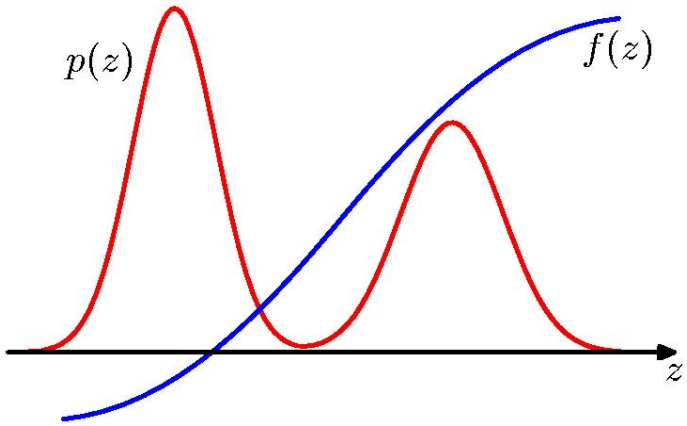
$$p(t_{new} \mid x_{new}, \mathbf{t}, \mathbf{X}, \theta) = E_{p(\mathbf{w}|\mathbf{t},\mathbf{X},\theta)}\left[p(t_{new} \mid x_{new}, \mathbf{w})\right] \approx \frac{1}{N_s}\sum_{s=1}^{N_s} p(t_{new} \mid x_{new}, \mathbf{w}_s)$$

- If needed, we can then compute such things as the mean and variance of this approximation to the predictive distribution.

Estimating expectations using Monte Carlo integration

- We wish to estimate expectations

$$E[f] = \int f(\mathbf{z}) p(\mathbf{z}) d\mathbf{z}$$



- Draw *L* samples *independently* from $p(\mathbf{z})$ and approximate the expectation with

$$\hat{f} = \frac{1}{L} \sum_{l=1}^{L} f(\mathbf{z}^{(l)}) \approx E[f]$$

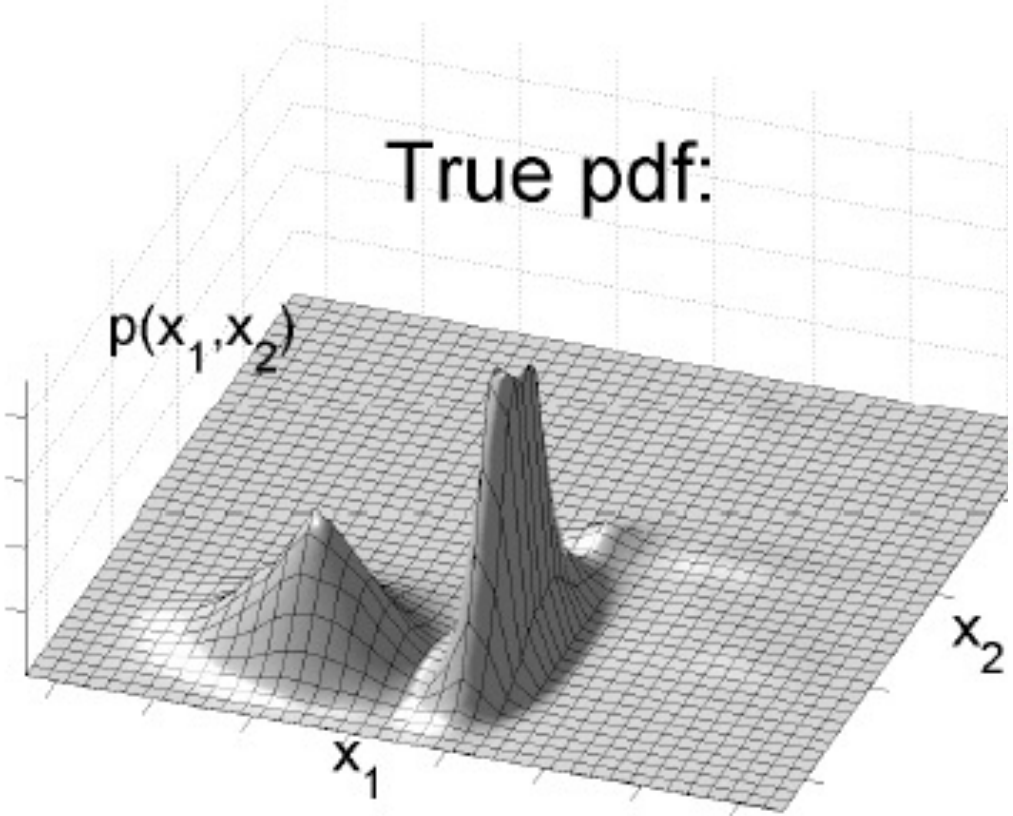
This approach is called *Monte Carlo integration*.

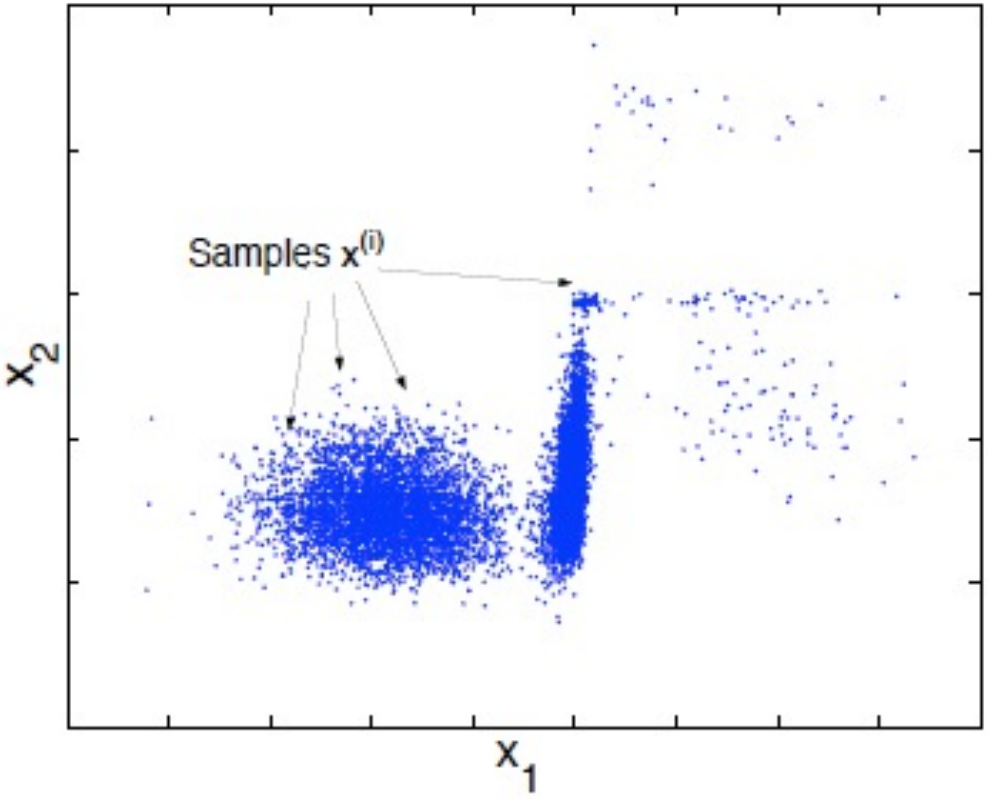
- The samples represent the distribution and in the limit

$$\lim_{L \to \infty} \hat{f} = E[f]$$ (rewrite of the Law of large numbers)

The estimator variance is $\mathrm{Var}[\hat{f}] = \mathrm{Var}[f]/L$

An example





Observation: Clearly for this complex distribution we need a lot of samples *L*

MAximum Posterior (MAP) Estimation

(Rogers & Girolami, Ch. 4.3)

Maximum Posterior (MAP) point estimate

- Find a parameter vector $\hat{\mathbf{w}}$ that maximizes the posterior density $p(\mathbf{w}|\mathbf{t}, \mathbf{X}, \theta)$: $\hat{\mathbf{w}} = \text{argmax}_{\mathbf{w}} p(\mathbf{w}|\mathbf{t}, \mathbf{X}, \theta)$.
- We do not need to know the marginal likelihood $p(\mathbf{t}|\mathbf{X})$ since it is a constant that does not change the maximum. We therefore only need the product of likelihood and prior, $g(\mathbf{w}, \mathbf{t}, \mathbf{X}, \theta) = p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\theta)$.
- As in MLE, we can take to logarithm to simplify the mathematical derivation and consider $\log g(\mathbf{w}, \mathbf{t}, \mathbf{X}, \theta)$.
- In general, we cannot perform analytical derivation of the maximum.
- We need a numerical optimization approach.

The Newton-Raphson method
Gradient ascent optimization

- The Newton-Raphson method is a gradient-based iterative optimization approach that searches for zero-crossings of the gradient function.
- Iteration step:

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \left(\left. \frac{\partial^2 \log g(\mathbf{w}, \mathbf{t}, \mathbf{X}, \theta)}{\partial \mathbf{w} \partial \mathbf{w}^T} \right|_{\mathbf{w}_i} \right)^{-1} \left. \frac{\partial \log g(\mathbf{w}, \mathbf{t}, \mathbf{X}, \theta)}{\partial \mathbf{w}} \right|_{\mathbf{w}_i}$$

Inverse Hessian matrix evaluated at $\mathbf{w}_i$

Gradient vector evaluated at $\mathbf{w}_i$

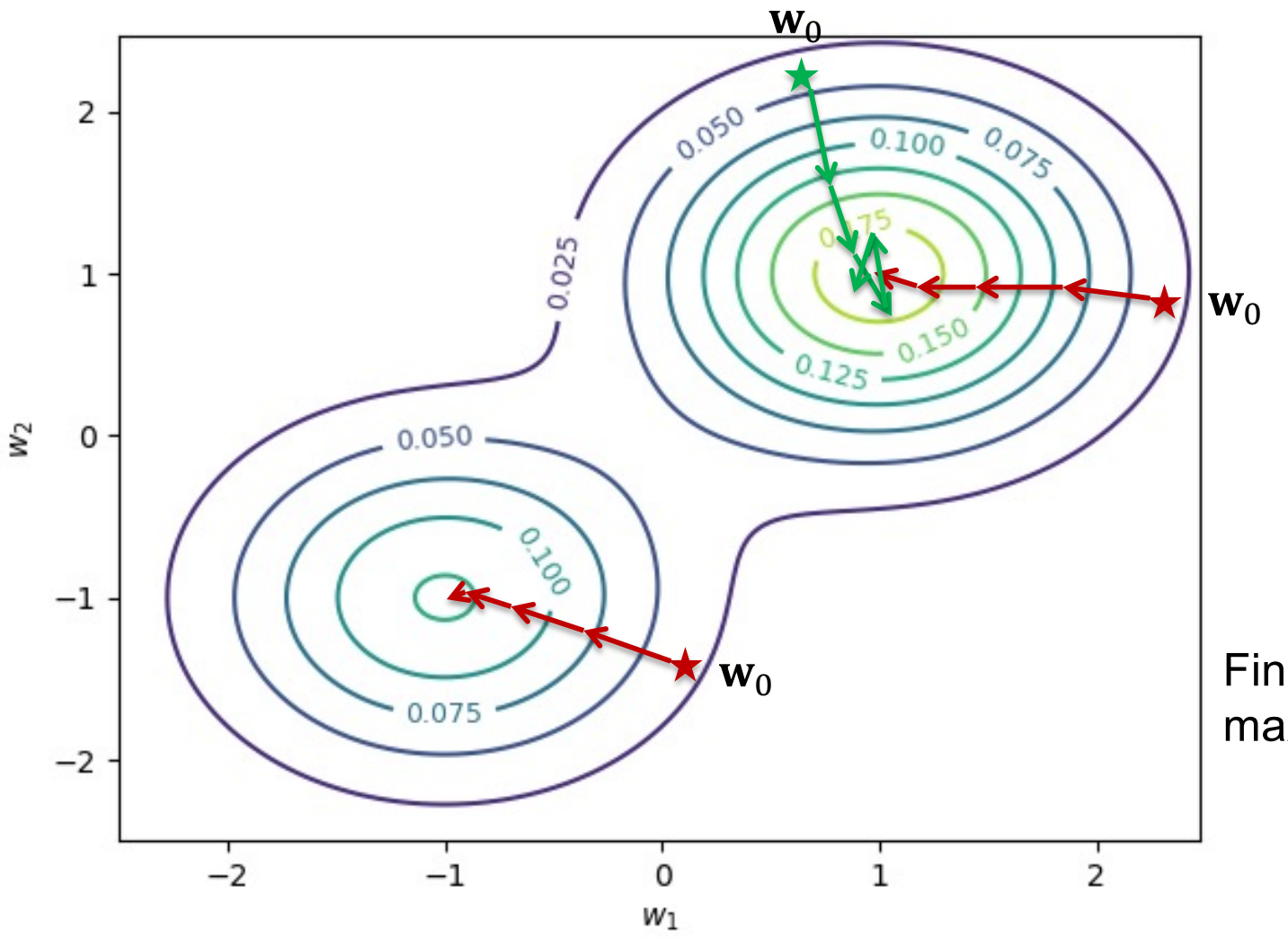
- We need an initial guess $\mathbf{w}_0$ to start the method.
- Iterate until some stoping criterium is met, e.g. the parameter update is very small (close to machine precision) or we have taken a fixed number of iteration steps.

Illustration of gradient optimization

Probability iso-contour plot (height map)

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \nabla \log g(\mathbf{w}_i)$$

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \mathbf{H}^{-1} \nabla \log g(\mathbf{w}_i)$$



Finds local maxima

Using the point estimate

- We cannot compute the predictive distribution since we only have a point estimate, but we can make inference using the likelihood, $p(t_{new}|\mathbf{x}_{new}, \widehat{\mathbf{w}})$.
- We need to be able to compute analytical derivatives for this method to work.
- Take care! If the posterior $p(\mathbf{w}|\mathbf{t}, \mathbf{X}, \theta)$ have multiple modes, there is no guarantee we find the global maximum.
- See the binary response example in the book.

Basic Sampling Methods

(Rejection_sampling.pdf)

What can sampling be used for?

We can use sampling:

- To estimate expectation value with respect to a probability model, e.g. estimate the predictive distribution $p(t_{new}|x_{new}, \mathbf{t}, \mathbf{X}, \theta)$, or simply computing higher order moments of a distribution.
- In some advanced machine learning methods. Ex.:
  - Sequential Monte Carlo techniques such as particle filtering (an extension of sampling-importance-resampling).
- To be able to synthesize data for testing purposes.

Sampling from “simple” distributions

- Specialized algorithms for some common distributions exist. Ex.:
  - Uniform distribution
  - Gaussian distribution, e.g. the Box-Muller method

Random Number Generators

- True random number generators:
  - Roll a dice, flip a coin, roulette wheel, …
  - Measure random fluctuations at atomic level (e.g. radioactive decay)
  - Measure hard disk head activity, computer clock drift, …

- Pseudo random number generators:
  - Based on a deterministic algorithm that generates a sequence of seemingly random numbers.
  - E.g. Linear congruential generator: $X_{n+1} = \left(aX_n + c\right) \bmod m$
  - Problems: The sequence is finite and deterministic when you know the starting point X_0 , called the random seed.
  - Try out: $m = 5$, $a = 4$, $c = 2$, $X_0 = 0$

    $m = 2^{32}$, $a = 1664525$, $c = 1013904223$, $X_0 = 100$

Pseudo random number generators: Take care!

- Pseudo random number generators are available in most programming languages and as libraries.
- But quality may vary! We want long periods!
- If you want random, then choose seed “randomly”!
- A standard approach is to use the system time as seed.
- Pythons `random` package by defaults set the seed using the system time.
- For debugging purpose you can fix the seed:
  - E.g. `random.seed(0)`
  - Consequence: You always get the same sequence of random numbers when you restart your program! (Try it out)

Sampling from “simple” distributions

- Specialized algorithms for some common distributions exist. Ex.:
  - Uniform distribution
  - Gaussian distribution, e.g. the Box-Muller method

- If we know the analytical expression for $p(\mathbf{z})$ - use the transformation method.

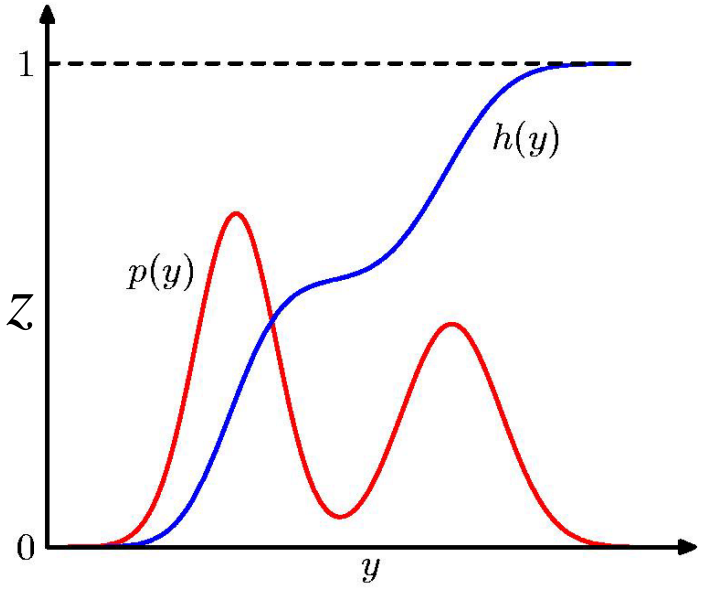
The transformation method

We want to sample from $p(y)$:

- Define the cumulative distribution function (CDF)

$$z = h(y) \equiv \int_{-\infty}^{y} p(\hat{y})d\hat{y}$$

Sample z from the uniform distribution $U(z|0,1)$ and apply $y = h^{-1}(z)$ which is distributed as $p(y)$.



- Requires that $p(y)$ is normalized and $h(y)$ is invertible.

Sampling from “simple” distributions

- Specialized algorithms for some common distributions exist. Ex.:
  - Uniform distribution
  - Gaussian distribution, e.g. the Box-Muller method

- If we know the analytical expression for $p(\mathbf{z})$ - use the transformation method.

- But what if our distribution is not “simple” and we cannot apply the transformation method?

Introducing proposal distributions

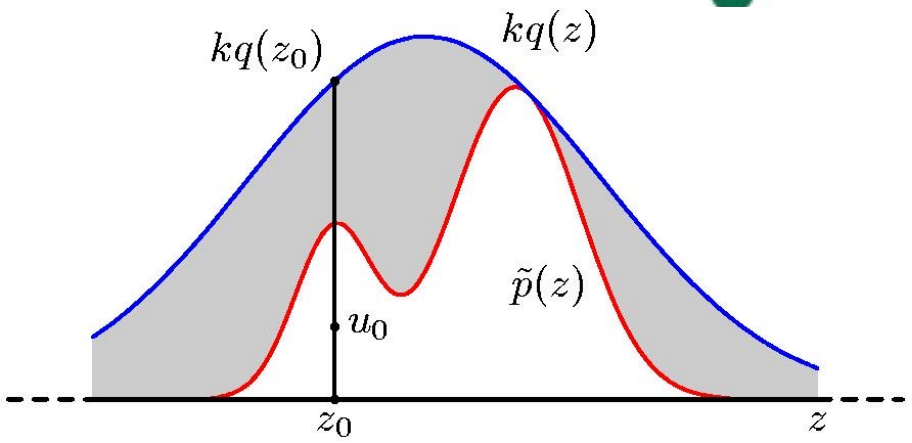
- Consider the distribution: $p(\mathbf{z}) = \tilde{p}(\mathbf{z})/Z_p$

- It may be difficult to sample from the distribution $p(\mathbf{z})$.
- Often $Z_p = \int \tilde{p}(\mathbf{z})d\mathbf{z}$ is difficult to compute, but $\tilde{p}(\mathbf{z})$ may be evaluated for any $\mathbf{z}$.

- Common strategy (used in rejection sampling, importance sampling, Metropolis-Hastings, etc.):
  - Use a much simpler proposal distribution $q(z)$ from which we can sample.
  - Generate a proposal sample and evaluate an acceptance criterion for the sample.

Rejection sampling

- Choose constant *k* and proposal distribution so $kq(z) \geq \tilde{p}(z)$ for all z.
- Sample z_0 from $q(z)$

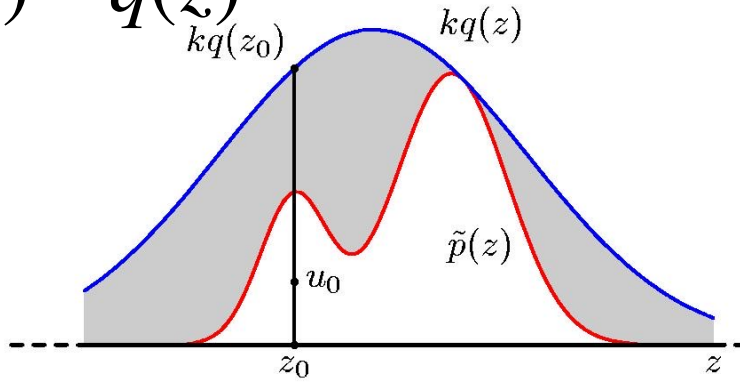


- Sample u_0 from $\mathcal{U}(u \mid 0, kq(z_0))$
- Reject z_0, if $u_0 > \tilde{p}(z_0)$ (in the gray area), otherwise keep z_0 (in the white area)

Assumption: The proposal distribution $q(z)$ must have a support larger than or equal to $p(z)$

Rejection sampling
Do we get the correct result?

- The distribution of a proposal is $\Pr(z) = q(z)$
- For a given sample z, the probability of acceptance is
  $\Pr(\text{accept} \mid z) = \tilde{p}(z)/kq(z)$
- The probability of acceptance is



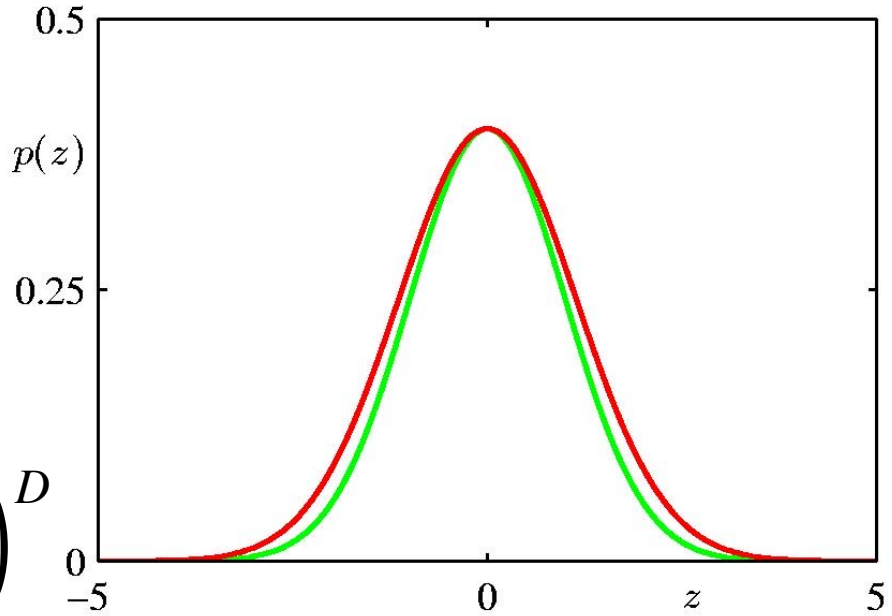
$$\Pr(\text{accept}) = \int \Pr(\text{accept} \mid z)\Pr(z)dz = \int \frac{\tilde{p}(z)}{kq(z)} q(z)dz$$
$$= \frac{1}{k}\int \tilde{p}(z)dz = Z_p/k$$

- The distribution of accepted samples is $\Pr(z \mid \text{accept}) =$
$$\frac{\Pr(\text{accept} \mid z)\Pr(z)}{\Pr(\text{accept})} = \frac{\{\tilde{p}(z)/kq(z)\}q(z)}{Z_p/k} = \tilde{p}(z)/Z_p = p(z)$$

Scalability of rejection sampling
What happens when we consider a D-dim z?

- Example: $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}\,|\,0,\sigma_p^2\mathbf{I})$

  $q(\mathbf{z}) = \mathcal{N}(\mathbf{z}\,|\,0,\sigma_q^2\mathbf{I})$

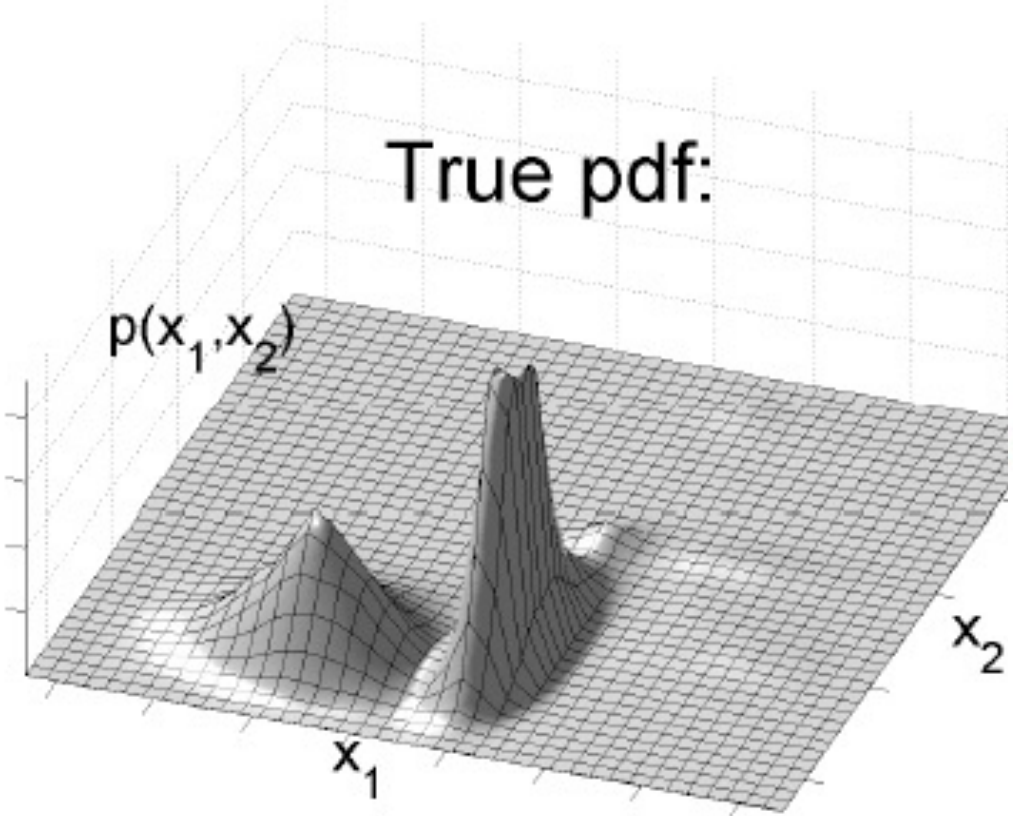


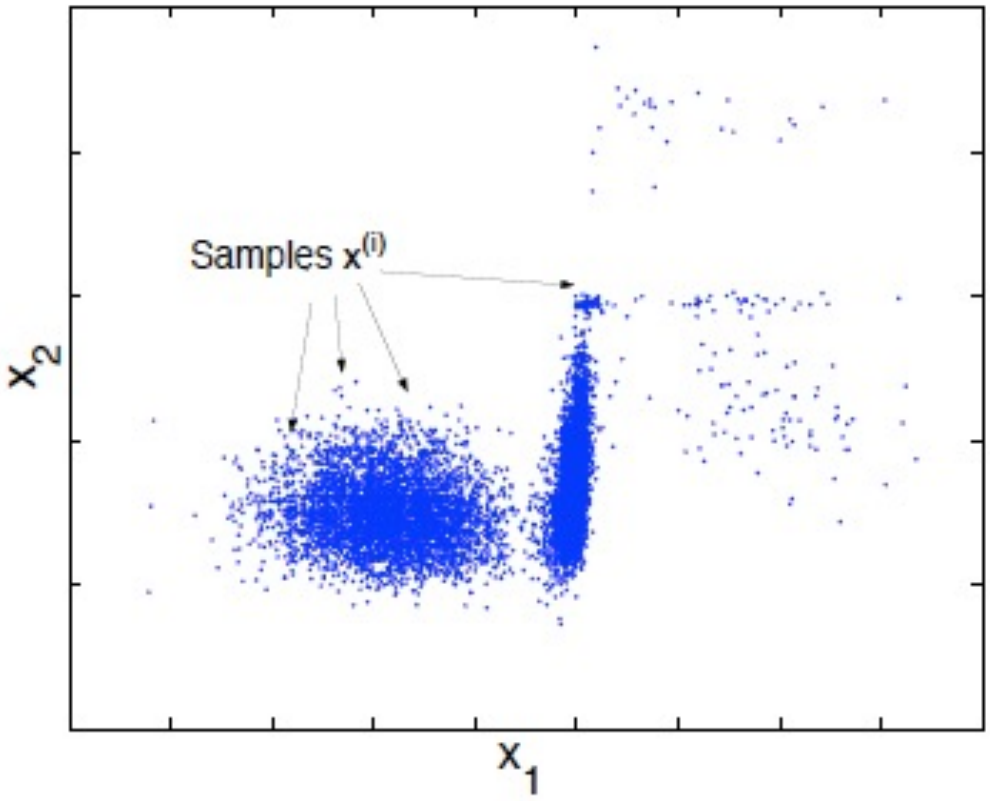
- Where $\sigma_q^2 \geq \sigma_p^2$ so $kq(\mathbf{z}) \geq p(\mathbf{z})$
- The optimal choice: $k = \left(\sigma_q / \sigma_p\right)^D$
- Hence the acceptance diminishes exponentially with dimensionality

$$\Pr(\text{accept}) = \frac{1}{k}\int p(z)dz = 1/k = \left(\sigma_q/\sigma_p\right)^{-D}$$

- Conclusion: As D grows more samples will be rejected, so rejection sampling will take more time to get *N* samples.

An example where rejection sampling is a poor fit





Sampling methods using Markov chains as proposal distribution

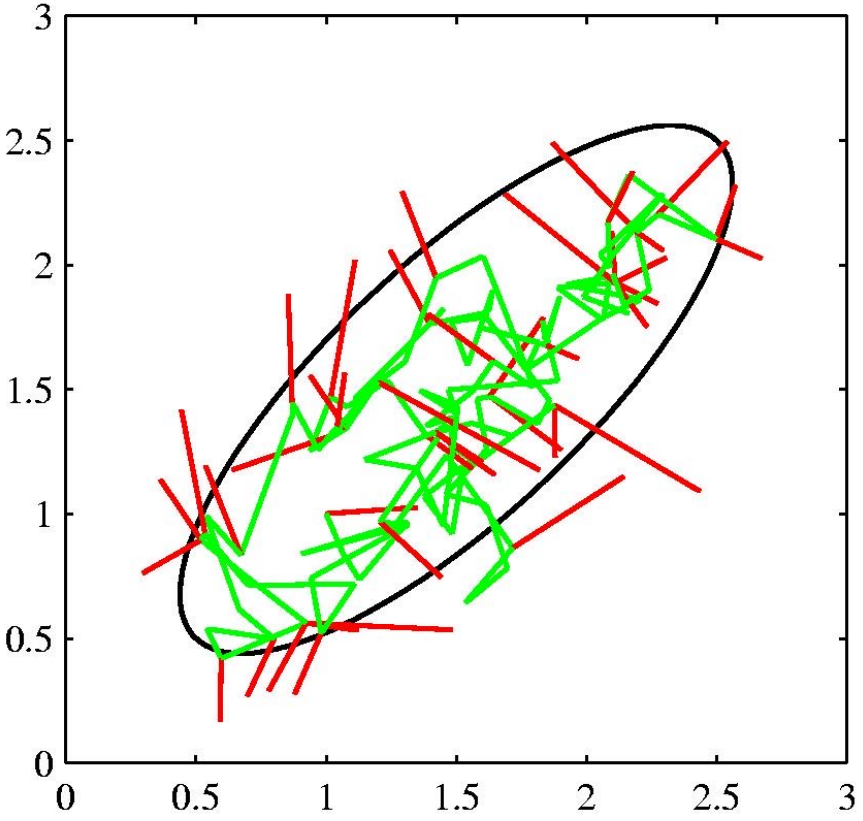
(Rogers & Girolami, Ch. 4.5)

Markov Chain Monte Carlo (MCMC) sampling

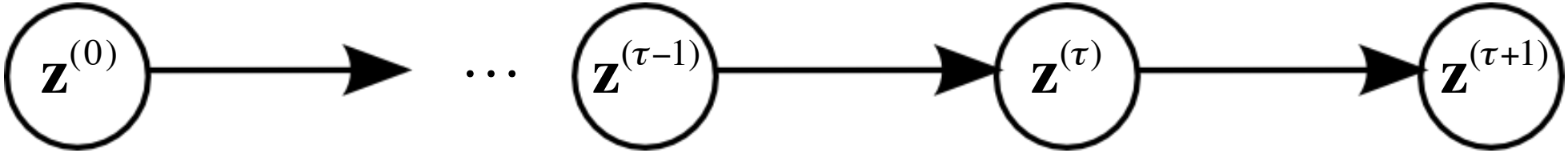
- Let the proposal distribution be dependent on the current state $\mathbf{z}^{(\tau)}$, $q(\mathbf{z} \mid \mathbf{z}^{(\tau)})$.
- Samples $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \mathbf{z}^{(3)}, \ldots$ form a 1st order Markov chain with

$$q(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(\tau)}) = q(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)})$$

- MCMC methods explore state space by a random walk.
- Scales well with dimensionality.
- Consequence: Samples are not independent, but we can select a subset of samples that are independent.



Markov chains



- A discrete time stochastic process is a first order Markov chain (MC), if

$$p(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)}, \mathbf{z}^{(\tau-1)}, \ldots, \mathbf{z}^{(0)}) = p_{\tau}(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)}) \quad \text{(Markov property)}$$

- Its Homogeneous (stationary), if $p_{\tau}(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)}) = p(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)})$
- Its fully specified by $p(\mathbf{z}^{(0)})$ and transition $p(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)})$
- Taking τ steps:

$$p(\mathbf{z}^{(0)}, \ldots, \mathbf{z}^{(\tau)}) = p(\mathbf{z}^{(0)}) \prod_{n=1}^{\tau} p(\mathbf{z}^{(n)} \mid \mathbf{z}^{(n-1)})$$

What properties would we like of the Markov chain?

- For any choice of distribution $q(\mathbf{z}^{(0)})$ of the starting point $\mathbf{z}^{(0)}$, we want the sample distribution to converge to the distribution $p(\mathbf{z})$ as $\tau \rightarrow \infty$.

- We also want to make sure that we can visit all parts of state space where $p(\mathbf{z}) > 0$. Otherwise, MCMC will not be able to produce samples distributed according to $p(\mathbf{z})$.

The Metropolis-Hastings sampler
A MCMC method for sampling from $p(\mathbf{z}) = \tilde{p}(\mathbf{z}) / Z_p$

1. Pick an initial sample state $\mathbf{z}^{(0)}$ for the Markov chain
2. Generate a sample $\mathbf{z}^*$ from the proposal distribution $q(\mathbf{z} \mid \mathbf{z}^{(\tau)})$
3. Evaluate the acceptance probability ratio

$$A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\mathbf{z}^*) q(\mathbf{z}^{(\tau)} \mid \mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)}) q(\mathbf{z}^* \mid \mathbf{z}^{(\tau)})}\right)$$

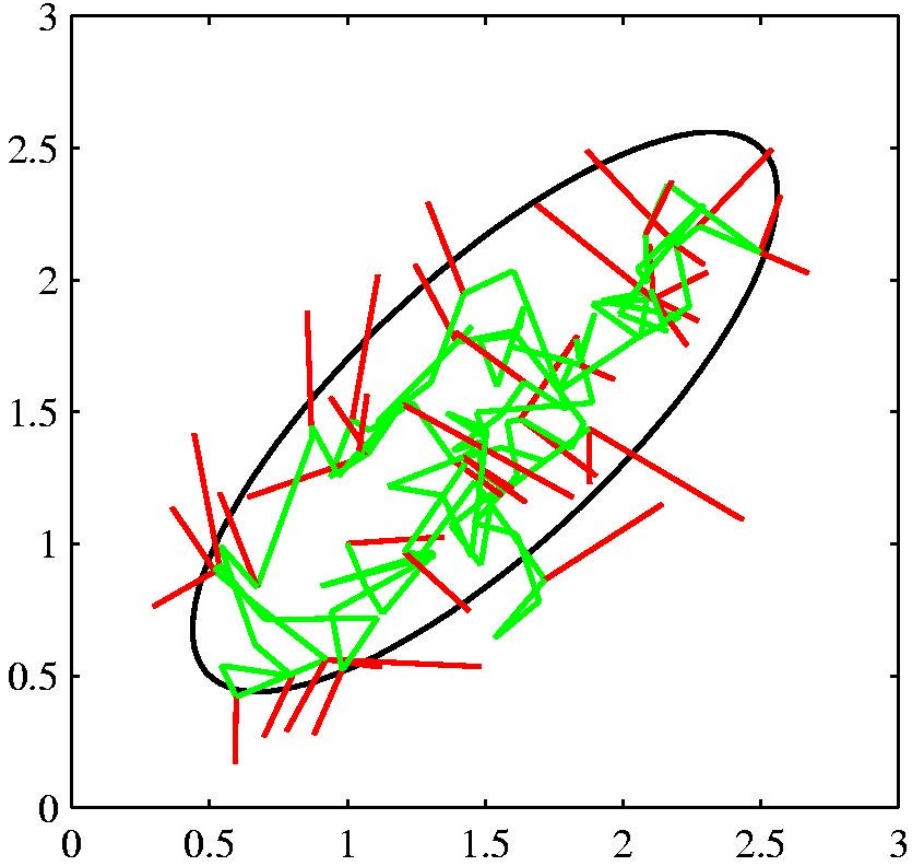
4. Pick a uniform random number, $u \sim \mathcal{U}(u \mid 0,1)$
5. Accept the sample $\mathbf{z}^*$ if $A(z^*, z^{(\tau)}) \geq u$ and set $\mathbf{z}^{(\tau+1)} = \mathbf{z}^*$, otherwise reuse current sample and set $\mathbf{z}^{(\tau+1)} = \mathbf{z}^{(\tau)}$
6. Repeat 2.) for $q(\mathbf{z} \mid \mathbf{z}^{(\tau+1)})$

The book presents another formulation in Fig. 4.10, but it is identical to this formulation.

The Metropolis-Hastings applied to a Gaussian

$$p(\mathbf{z}) = \mathcal{N}\left(\mathbf{z} \mid \mu, \Sigma\right)$$

$$q(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)}) = \mathcal{N}\left(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)}, \rho^2 = 0.2^2\mathbf{I}\right)$$



Metropolis is a special case of the M-H sampler

- Assuming the proposal distribution is symmetric

  $$q(\mathbf{z}_A \mid \mathbf{z}_B) = q(\mathbf{z}_B \mid \mathbf{z}_A)$$

  we get the simplified acceptance criterion

  $$A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)})}\right)$$

- This is also know as the Metropolis sampler
- An example is to use a Gaussian distribution as the proposal distribution (it is symmetric).

Sampling for Bayesian inference

- For general Bayesian models we can approximate the expectation in the expression for the predictive distribution by sampling N_s samples of parameter vectors $\mathbf{w}_s$ from the posterior and use this approximation

$$p(t_{new} \mid x_{new}, \mathbf{t}, \mathbf{X}, \theta) = E_{p(\mathbf{w}|\mathbf{t},\mathbf{X},\theta)}\left[p(t_{new} \mid x_{new}, \mathbf{w})\right] \approx \frac{1}{N_s}\sum_{s=1}^{N_s} p(t_{new} \mid x_{new}, \mathbf{w}_s)$$

- Using the Metropolis-Hastings sampler, we just need to be able to evaluate $\tilde{p}(\mathbf{w}|\mathbf{t}, \mathbf{X}, \theta) = p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\theta)$ and pick a prober proposal distribution $q(\mathbf{w}|\mathbf{w}^{(\tau)})$.

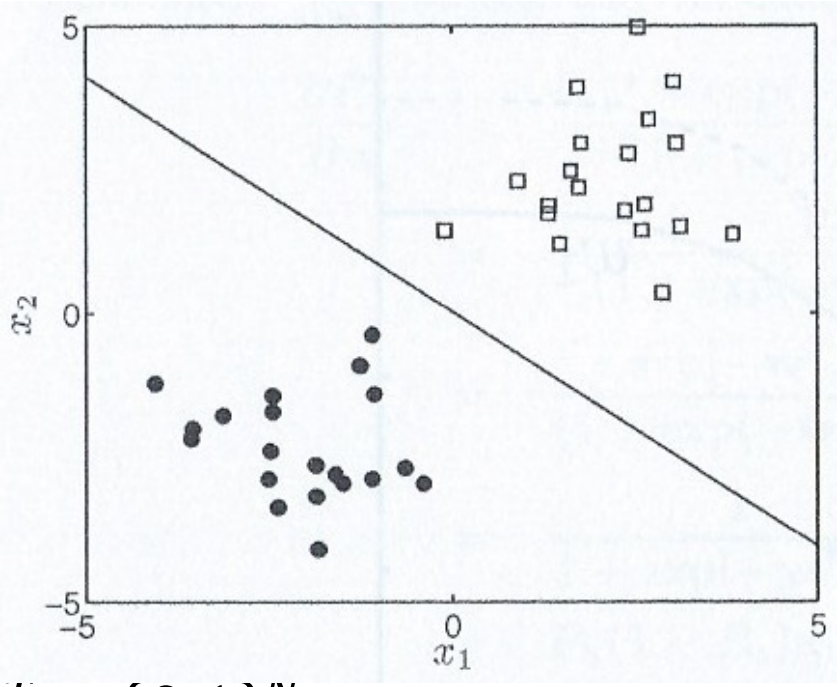
The Binary Response Model Example

(Rogers & Girolami, Ch. 4.2)

Example: Bayesian binary classification model

- The model, $\mathbf{w}^T\mathbf{x} = 0$, represents a linear decision boundary – all data on the same side of the boundary belongs to the same class.
- Class labels: $t \in \{0,1\}$
- Input vector: $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2$
- Parameters: $\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \in \mathbb{R}^2$
- Training data:

$$\mathbf{X} = [\mathbf{x}_1 \quad \cdots \quad \mathbf{x}_N]^T$$
$$\mathbf{t} = [t_1 \quad \cdots \quad t_N]^T \in \{0,1\}^N$$



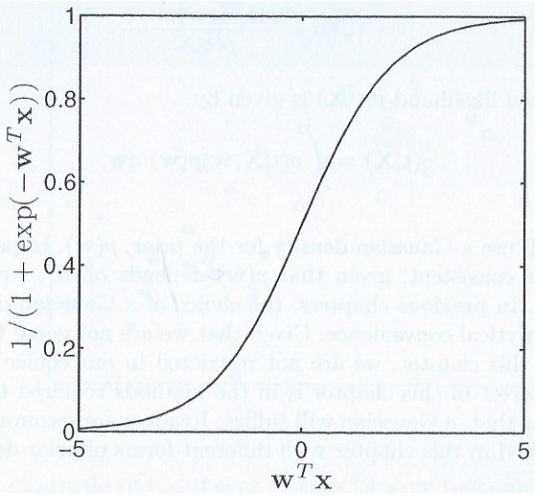
Which class does $\mathbf{x}$ belong to?

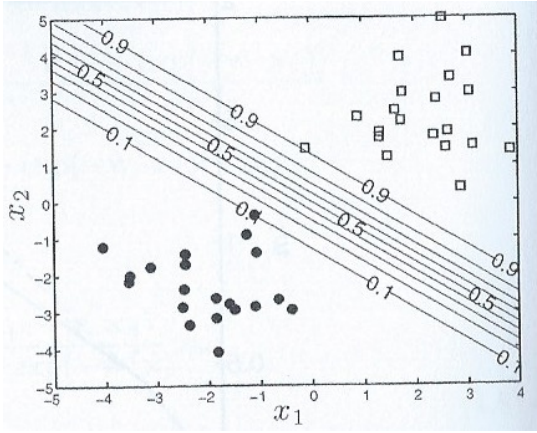
- Instead of assigning the class label based on which side of the line, $\mathbf{w}^T\mathbf{x} = 0$, the input $\mathbf{x}$ lies, we instroduce a probability model:
- Consider the class as a binary random variable T with probability distribution (class probability) given by

$$P(T = 1|\mathbf{x}, \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^T\mathbf{x})}$$

and

$$P(T = 0|\mathbf{x}, \mathbf{w}) = 1 - P(T = 1|\mathbf{x}, \mathbf{w}) = \frac{\exp(-\mathbf{w}^T\mathbf{x})}{1 + \exp(-\mathbf{w}^T\mathbf{x})}$$





Bayesian model

- $p(\mathbf{w}|\mathbf{t}, \mathbf{X}, \theta) = \frac{p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\theta)}{p(\mathbf{t}|\mathbf{X})}$.
- Prior: $p(\mathbf{w}|\theta = \sigma^2) = \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$
- Likelihood (assuming i.i.d. training data):

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} P(T_n = 1|\mathbf{x}_n, \mathbf{w})^{t_n} P(T_n = 0|\mathbf{x}_n, \mathbf{w})^{1-t_n}$$

- The marginal likelihood $p(\mathbf{t}|\mathbf{X})$ is difficult to compute, so we have the unnormalized posterior

$$\tilde{p}(\mathbf{w}|\mathbf{t}, \mathbf{X}, \sigma^2) = p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\sigma^2)$$

- We can perform inference with this model by using MAP point estimates or by sampling from the posterior distribution.

Sampling From The Binary Response Model

(Rogers & Girolami, Ch. 4.5)

Setting up the Metropolis-Hastings sampler

- Prior: $p(\mathbf{w}|\theta = \sigma^2) = \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$
- Likelihood (assuming i.i.d. training data):

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} P(T_n = 1|\mathbf{x}_n, \mathbf{w})^{t_n} P(T_n = 0|\mathbf{x}_n, \mathbf{w})^{1-t_n}$$

- The marginal likelihood $p(\mathbf{t}|\mathbf{X})$ is difficult to compute, so we have the unnormalized posterior

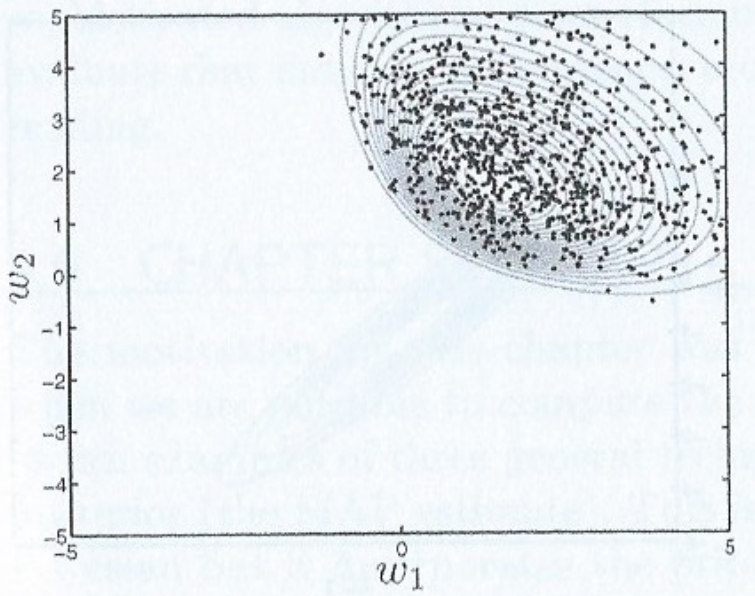
$$\tilde{p}(\mathbf{w}|\mathbf{t}, \mathbf{X}, \sigma^2) = p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\sigma^2)$$

- Choosing a proposal distribution:

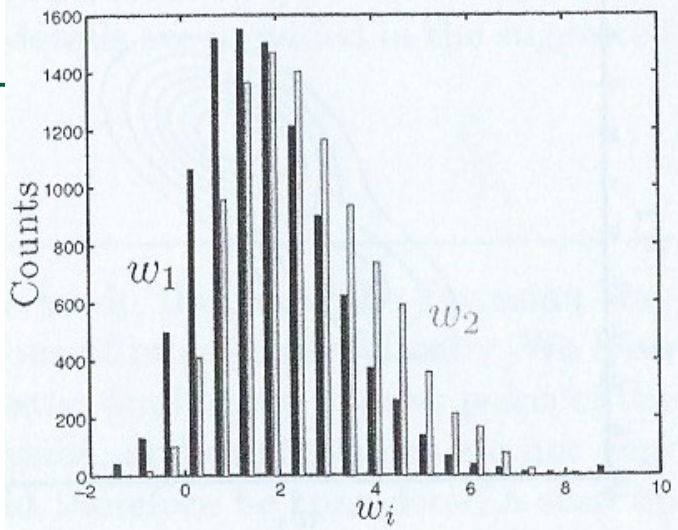
$$q(\tilde{\mathbf{w}}_s|\mathbf{w}_{s-1}, \boldsymbol{\Sigma}) = \mathcal{N}(\mathbf{w}_{s-1}, 0.5\mathbf{I})$$

- Choosing an initial sample, e.g. sample one from the prior distribution.

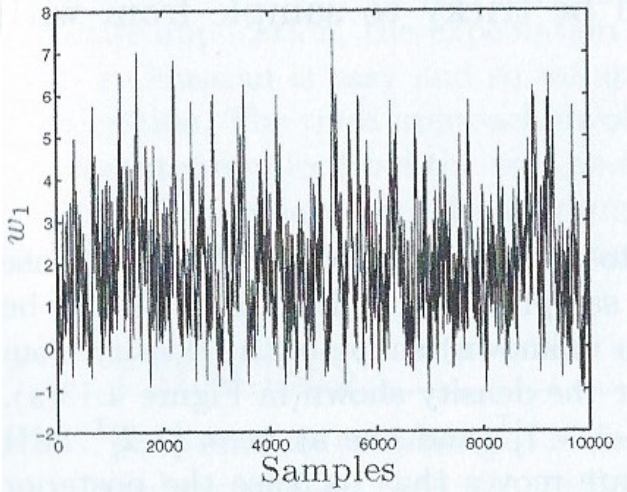
Example: Posterior samples in parameter space



(a) One thousand of the MH samples along with the posterior contours.



(b) Histograms of the samples for both w_1 (black) and w_2 (grey).



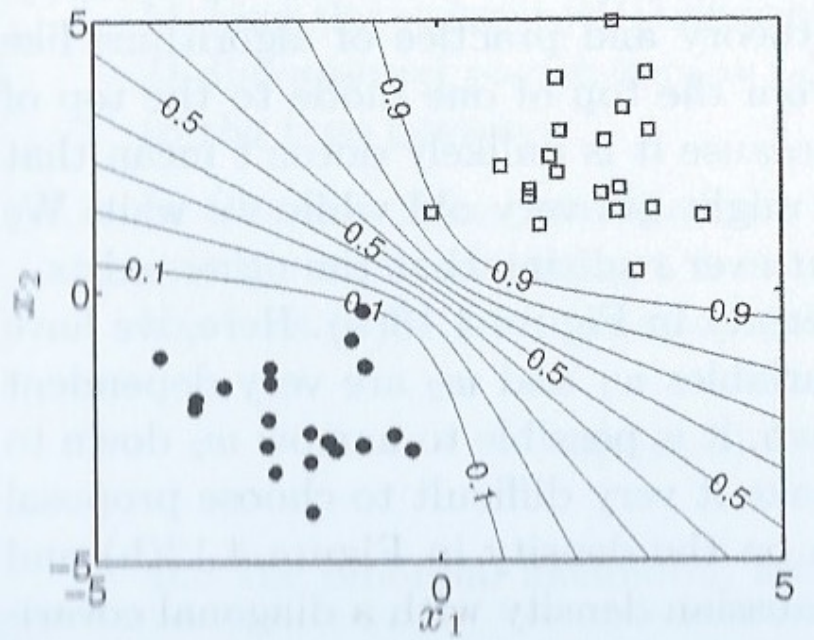
(c) All of the w_1 samples plotted against iteration, s.

Example: Bayesian Binary Classification

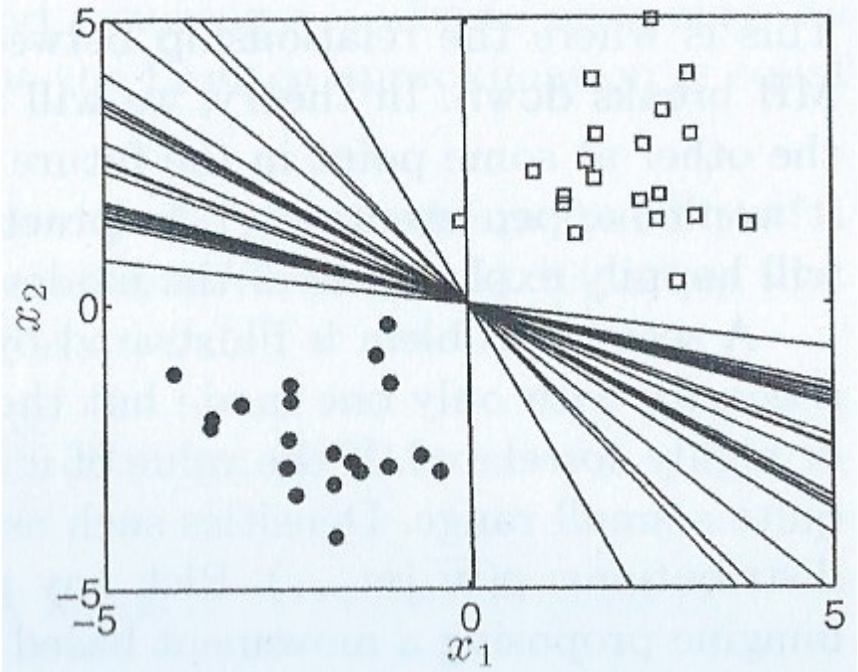
Approximate predictive distribution by Monte Carlo integration:

$$P(T_{new} = 1|\mathbf{x}_{new}, \mathbf{t}, \mathbf{X}, \sigma^2) = E_{p(\mathbf{w}|\mathbf{t},\mathbf{X},\sigma^2)}[P(T_{new} = 1|\mathbf{x}_{new}, \mathbf{w})]$$

$$\approx \frac{1}{M}\sum_{s=1}^{M} \frac{1}{1 + \exp(-\mathbf{w}_s^T \mathbf{x})}$$



(e) Predictive probability contours. The contours show the probability of classifying an object at any location as a square.

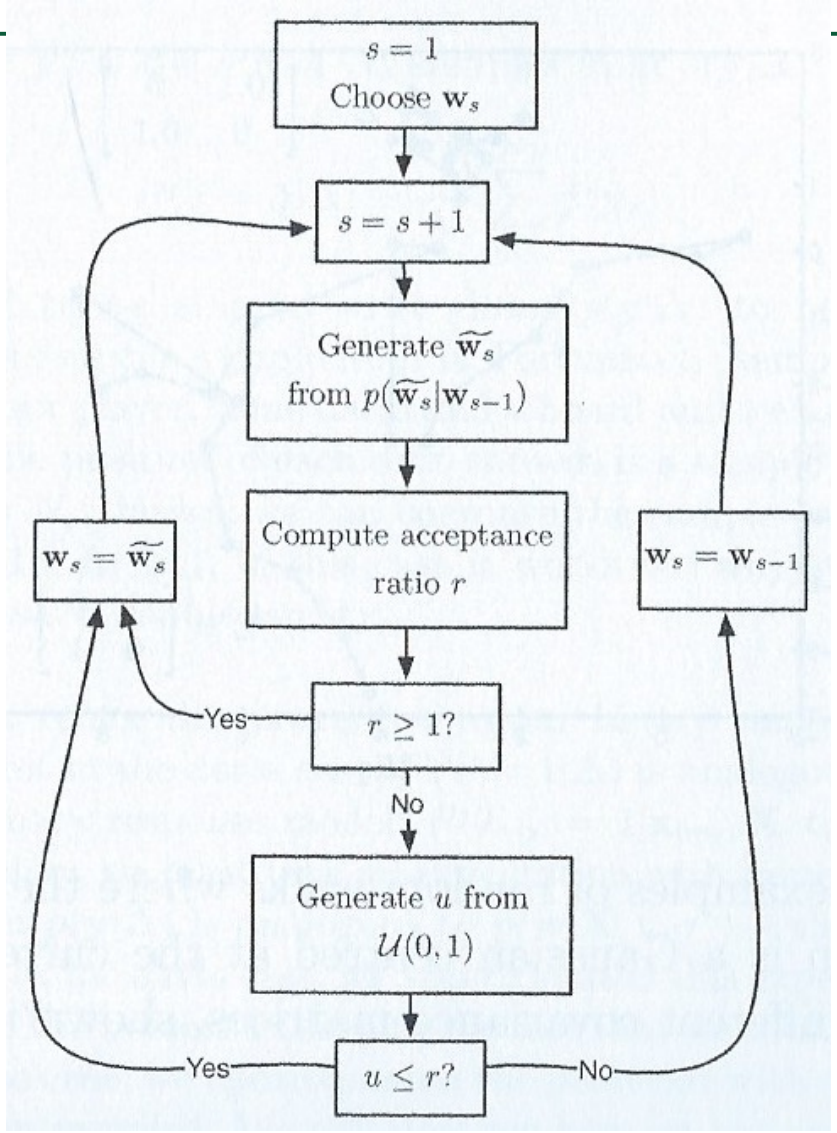


(f) Decision boundaries created from 20 randomly selected MH samples.

Why does Metropolis-Hastings work?

(Rogers & Girolami, Ch. 9.3)

The Metropolis-Hastings sampler
R&G's version, Fig. 4.10, page 156



Acceptance ratio:

$$r = \frac{\tilde{p}(\widetilde{\mathbf{w}}_s|\mathbf{t}, \mathbf{X}, \sigma^2)}{\tilde{p}(\mathbf{w}_{s-1}|\mathbf{t}, \mathbf{X}, \sigma^2)} \frac{q(\mathbf{w}_{s-1}|\widetilde{\mathbf{w}}_s)}{q(\widetilde{\mathbf{w}}_s|\mathbf{w}_{s-1})}$$

In the book, the proposal distribution is chosen to be:

$$q(\mathbf{w}_s|\mathbf{w}_{s-1}) = \mathcal{N}(\mathbf{w}_{s-1}, \boldsymbol{\Sigma})$$

Which is symmetric and

$$q(\widetilde{\mathbf{w}}_s|\mathbf{w}_{s-1}) = q(\mathbf{w}_{s-1}|\widetilde{\mathbf{w}}_s)$$

Therefore

$$r = \frac{\tilde{p}(\widetilde{\mathbf{w}}_s|\mathbf{t}, \mathbf{X}, \sigma^2)}{\tilde{p}(\mathbf{w}_{s-1}|\mathbf{t}, \mathbf{X}, \sigma^2)}$$

The Metropolis-Hastings (MH) sampler is a Markov chain

- Lets consider the general case of sampling from a distribution $p(\mathbf{z})$ using some proposal distribution $q(\mathbf{z}_{\tau+1}|\mathbf{z}_\tau)$. We refer to $\mathbf{z}$ as states or points.
- The probability that we accept a proposed sample $\mathbf{z}^*$ is given by

$$\mathrm{P}(\mathrm{accept}|\mathbf{z}^*) = A(\mathbf{z}^*, \mathbf{z}_\tau) = \min\left(1, \frac{p(\mathbf{z}^*)q(\mathbf{z}_\tau|\mathbf{z}^*)}{p(\mathbf{z}_\tau)q(\mathbf{z}^*|\mathbf{z}_\tau)}\right)$$

- The probability of transitioning from $\mathbf{z}_\tau$ to $\mathbf{z}_{\tau+1}$ in the Markov chain formed by the MH sampler is

$$T(\mathbf{z}_{\tau+1}, \mathbf{z}_\tau) = q(\mathbf{z}_{\tau+1}|\mathbf{z}_\tau)A(\mathbf{z}_{\tau+1}, \mathbf{z}_\tau)$$

Probability that we change to new sample $\mathbf{z}_{\tau+1}$

Probability density that we propose $\mathbf{z}_{\tau+1}$

Probability of accepting the proposed sample $\mathbf{z}_{\tau+1}$

What properties would we like of the Markov chain?

- For any choice of initial distribution $q(\mathbf{z}_0)$ of the starting point $\mathbf{z}_0$, we want the sample distribution to converge to the distribution $p(\mathbf{z})$ as we get more samples, ideally as $\tau \to \infty$.

- We also want to make sure that we can visit all parts of $\mathbf{z}$ space for which $p(\mathbf{z}) > 0$. Otherwise, MCMC will not be able to produce samples distributed according to $p(\mathbf{z})$.

Properties of Markov chains

- *Transition probability* for a *homogeneous Markov chain*

$$q(\mathbf{z}_{\tau+1}|\mathbf{z}_\tau)$$

- *Marginal* probability distribution after one step of the Markov chain:

$$p(\mathbf{z}_{\tau+1}) = \int q(\mathbf{z}_{\tau+1}|\mathbf{z}_\tau)p(\mathbf{z}_\tau)d\mathbf{z}_\tau$$

- For an *invariant distribution* $p^*(\mathbf{z})$ of a Markov chain it holds that:

$$p^*(\mathbf{z}_{\tau+1}) = \int q(\mathbf{z}_{\tau+1}|\mathbf{z}_\tau)p^*(\mathbf{z}_\tau)d\mathbf{z}_\tau$$

- A sufficient condition for ensuring $p^*(\mathbf{z})$ is an invariant distribution of the Markov chain is given by choosing $q(\mathbf{z}|\mathbf{z}')$ to satisfy the *detailed balance* equation $p^*(\mathbf{z})q(\mathbf{z}'|\mathbf{z}) = p^*(\mathbf{z}')q(\mathbf{z}|\mathbf{z}')$.
- If detailed balance is fulfilled, then

$$\int q(\mathbf{z}|\mathbf{z}')p^*(\mathbf{z}')d\mathbf{z}' = \int q(\mathbf{z}'|\mathbf{z})p^*(\mathbf{z})d\mathbf{z}' = p^*(\mathbf{z})\int q(\mathbf{z}'|\mathbf{z})d\mathbf{z}' = p^*(\mathbf{z})$$

What we need for a MCMC based sampler to work: Sampling from $p(\mathbf{z})$

- We want a Markov chain such that $p(\mathbf{z})$ is an invariant.
- We also require that the distribution of samples $p(\mathbf{z}_\tau)$ converge to $p(\mathbf{z})$ as $\tau \to \infty$, irrespectively of the choice of initial distribution $q(\mathbf{z}_0)$.
- Some initial burn-in samples are to be expected before convergence is achieved.
- Such a Markov chain is called an *ergodic* Markov chain.

- A sufficient condition for ergodicity is that the conditional distributions $q(\mathbf{z}_{\tau+1}|\mathbf{z}_\tau)$ are nowhere zero, $q(\mathbf{z}_{\tau+1}|\mathbf{z}_\tau) > 0$ for all $\mathbf{z}_{\tau+1}$, $\mathbf{z}_\tau$.
- This ensures that any state can be reached from any other state in a finite number of steps.

Convergence of The Metropolis-Hastings (MH) sampler Sampling from $p(\mathbf{z})$

- We just need to check that $p(\mathbf{z})$ is invariant with respect to the Markov chain formed by the MH sampler, by verifying that the transition probability

$$T(\mathbf{z}_{\tau+1}, \mathbf{z}_\tau) = q(\mathbf{z}_{\tau+1}|\mathbf{z}_\tau)A(\mathbf{z}_{\tau+1}, \mathbf{z}_\tau)$$

- satisfy the detailed balance equation

$$p(\mathbf{z})T(\mathbf{z}', \mathbf{z}) = p(\mathbf{z})q(\mathbf{z}'|\mathbf{z})A(\mathbf{z}', \mathbf{z}) = \min(p(\mathbf{z})q(\mathbf{z}'|\mathbf{z}), p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}'))$$

$$= \min\big(p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}'), p(\mathbf{z})q(\mathbf{z}'|\mathbf{z})\big) = p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}')A(\mathbf{z}, \mathbf{z}') = p(\mathbf{z}')T(\mathbf{z}, \mathbf{z}')$$

$$A(\mathbf{z}',\mathbf{z}) = \min\left(1, \frac{p(\mathbf{z}')q(\mathbf{z}\,|\,\mathbf{z}')}{p(\mathbf{z})q(\mathbf{z}'\,|\,\mathbf{z})}\right) \qquad A(\mathbf{z},\mathbf{z}') = \min\left(1, \frac{p(\mathbf{z})q(\mathbf{z}'\,|\,\mathbf{z})}{p(\mathbf{z}')q(\mathbf{z}\,|\,\mathbf{z}')}\right)$$

- However convergence rate and correlation between samples will dependent on the choice of proposal distribution.

Sampling problems

(Rogers & Girolami, Ch. 9.4)

The log-acceptance ratio version of the Metropolis-Hastings sampler

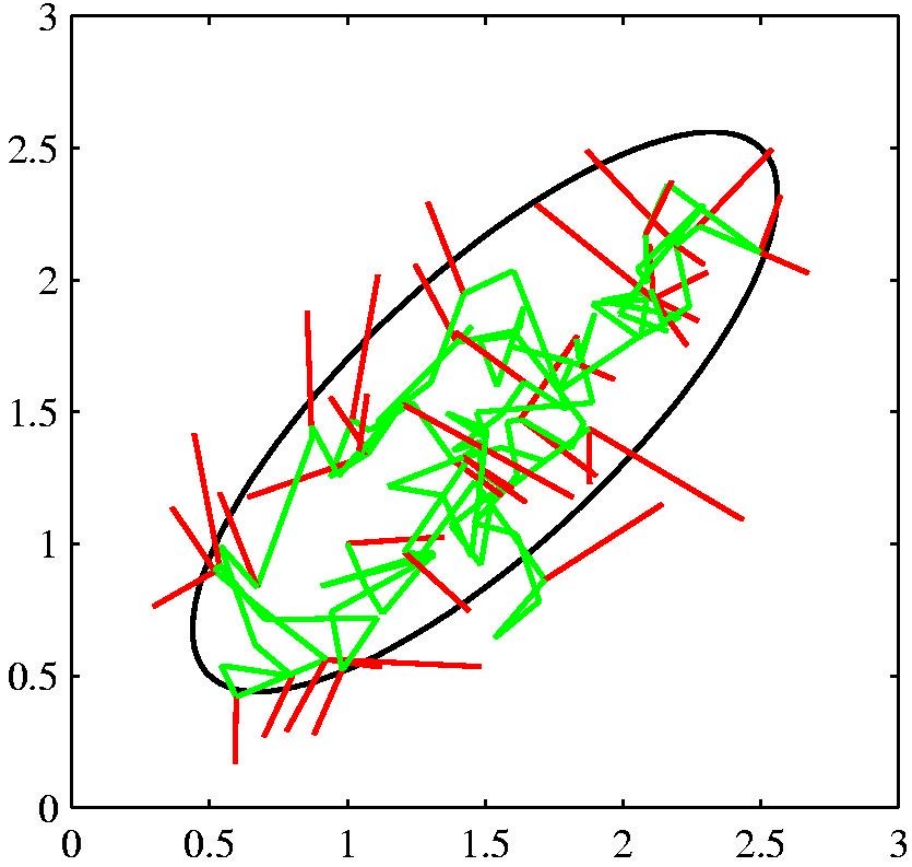
- For some choices of probability distributions (and training data) we might run into numerical problems when sampling using Metropolis-Hastings algorithm.
- In this case we can use the log-acceptance ratio formulation of the algorithm: $$\log A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = \min\left(0, \log\left(\frac{\tilde{p}(\mathbf{z}^*)q(\mathbf{z}^{(\tau)} \mid \mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)})q(\mathbf{z}^* \mid \mathbf{z}^{(\tau)})}\right)\right)$$
- This means we need to change step 3 in the algorithm as well a step 5 into

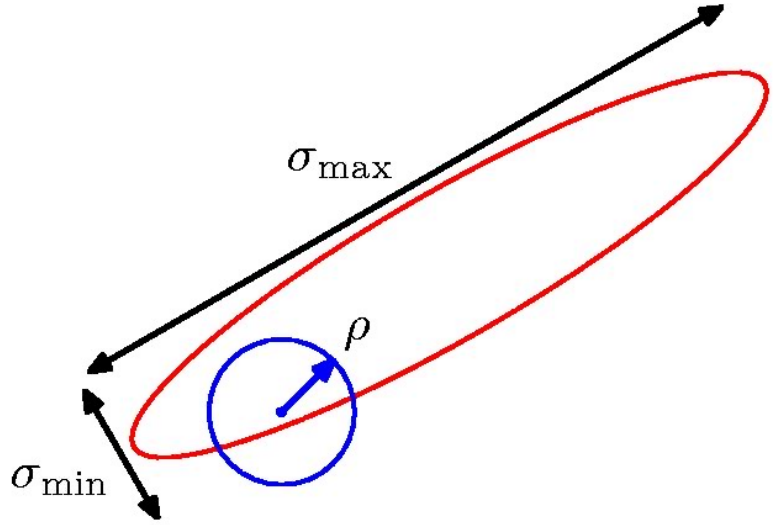
5. Accept the sample $\mathbf{z}^*$ if $\log A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) > \log u$ and set $\mathbf{z}^{(\tau+1)} = \mathbf{z}^*$, otherwise reuse current sample and set $\mathbf{z}^{(\tau+1)} = \mathbf{z}^{(\tau)}$

The Metropolis-Hastings applied to a Gaussian

$$p(\mathbf{z}) = \mathcal{N}\left(\mathbf{z} \mid \mu, \Sigma\right)$$

$$q(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)}) = \mathcal{N}\left(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)}, \rho^2 = 0.2^2\mathbf{I}\right)$$



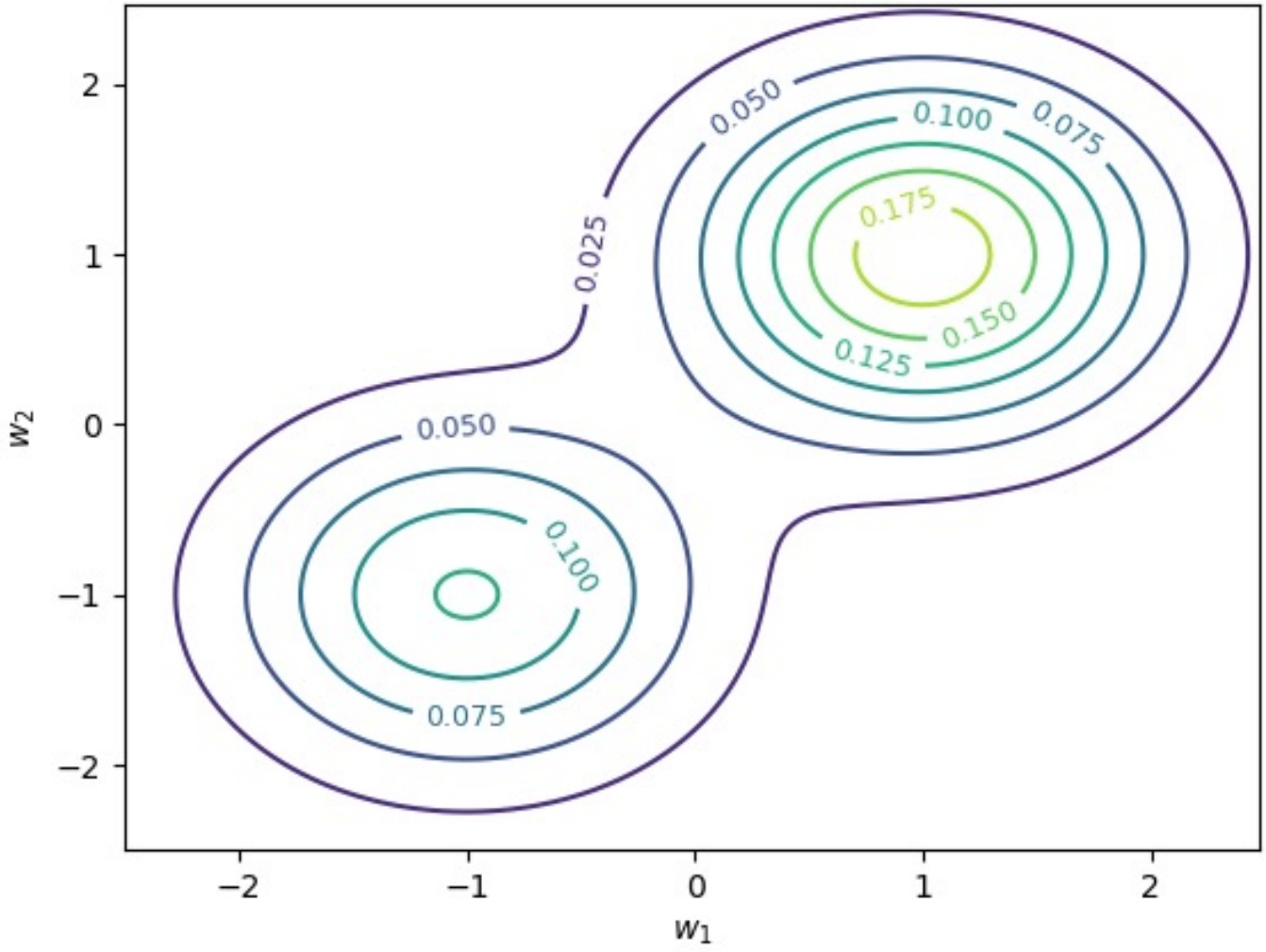


A good choice: $\rho = \sigma_{\min}$

Samples are independent approximately in order of steps: $\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^2$

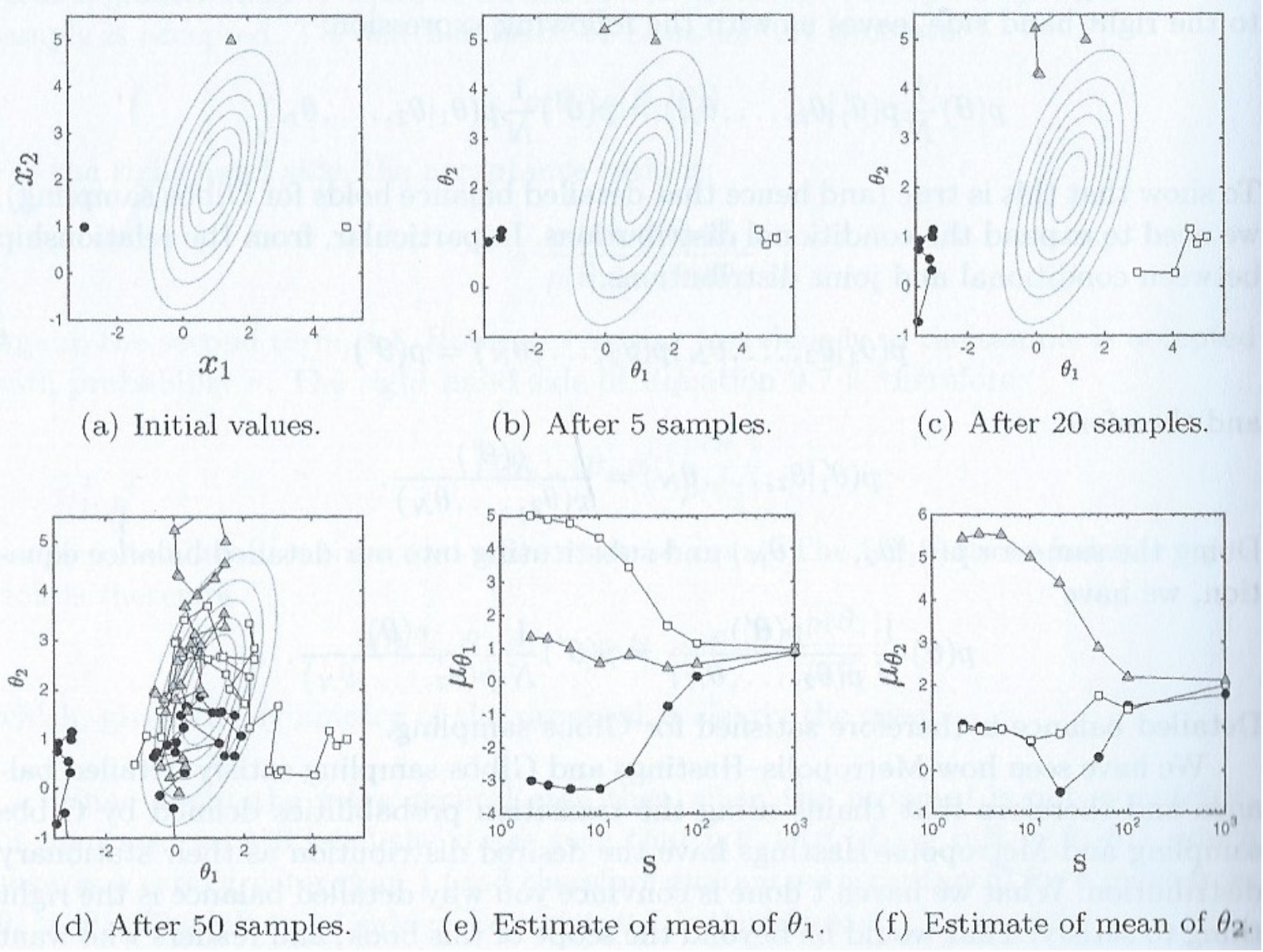
This type of distribution also pose a problem for the Metropolis-Hastings sampler

$$A(\mathbf{z}',\mathbf{z}) = \min\left(1, \frac{p(\mathbf{z}')q(\mathbf{z}\,|\,\mathbf{z}')}{p(\mathbf{z})q(\mathbf{z}'\,|\,\mathbf{z})}\right)$$



Burn-in samples needed for convergence to

Example: 3 MH samplers for a Gaussian distribution $p(\mathbf{z})$



(a) Initial values. (b) After 5 samples. (c) After 20 samples.

(d) After 50 samples. (e) Estimate of mean of θ_1. (f) Estimate of mean of θ_2.

Measuring correlation between samples

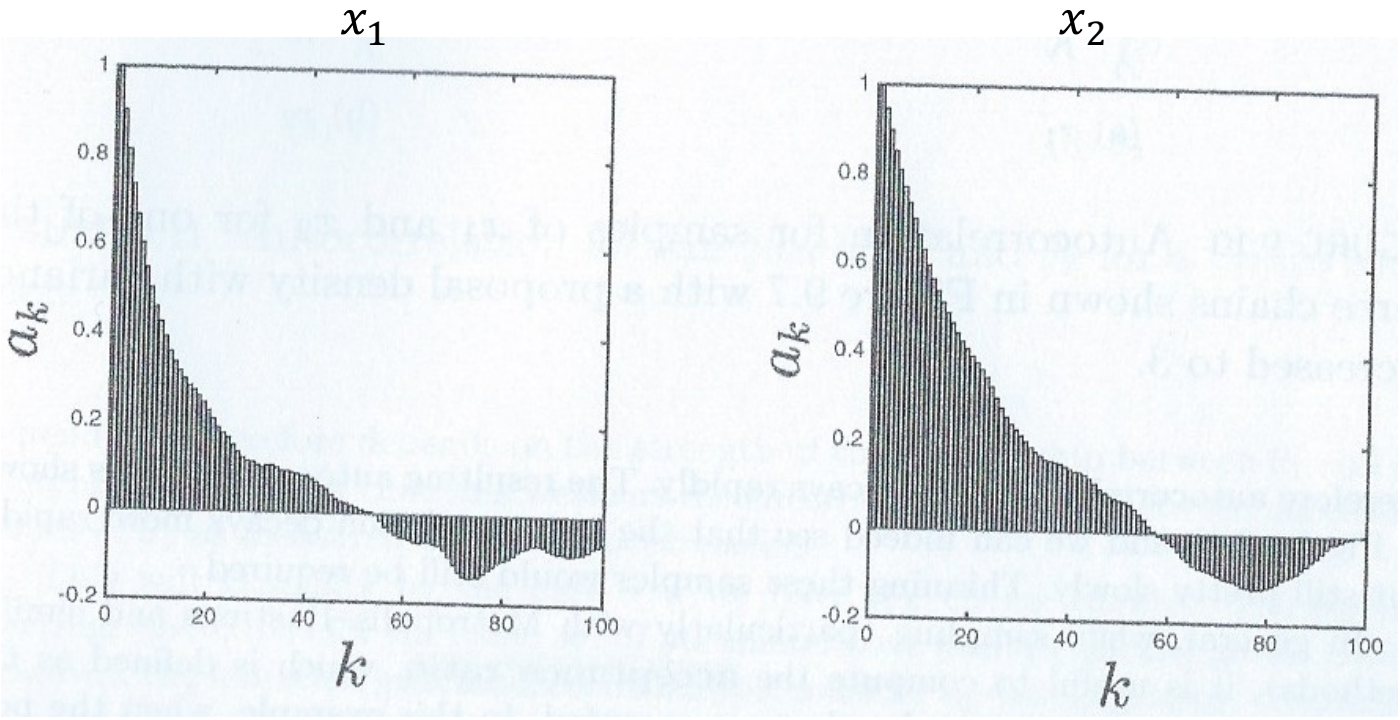
- 1-D autocorrelation with lag *k*=1, 2, …

$$\hat{R}(k) = \frac{1}{(L-k)\hat{\sigma}^2} \sum_{l=1}^{L-k} (y^{(l)} - \hat{\mu})(y^{(l+k)} - \hat{\mu})$$

- Sample mean $\hat{\mu}$ and sample variance $\hat{\sigma}^2$
- Perfect correlation/anti-correlation $\hat{R} = \pm 1$
- De-correlated when $\hat{R} = 0$
- De-correlated samples can be considered independent.

Visualizing autocorrelation

Autocorrelation for one sample sequence from previous example



Sampling for Bayesian inference

- For general Bayesian models we can approximate the expectation in the expression for the predictive distribution by sampling N_s samples of parameter vectors $\mathbf{w}_s$ from the posterior and use this approximation

$$p(t_{new} \mid x_{new}, \mathbf{t}, \mathbf{X}, \theta) = E_{p(\mathbf{w}|\mathbf{t},\mathbf{X},\theta)}\left[p(t_{new} \mid x_{new}, \mathbf{w})\right] \approx \frac{1}{N_s}\sum_{s=1}^{N_s} p(t_{new} \mid x_{new}, \mathbf{w}_s)$$

- We can use sampling algorithms such as Metropolis-Hastings sampler to generate the samples we need for this approximation.

Summary

- Basic sampling methods
  - Rejection sampling
- Sampling using Markov Chain Monte Carlo methods:
  - Metropolis-Hastings sampler
- The problem is that we do not get independent samples, i.e. the samples are correlated so we need to skip samples.

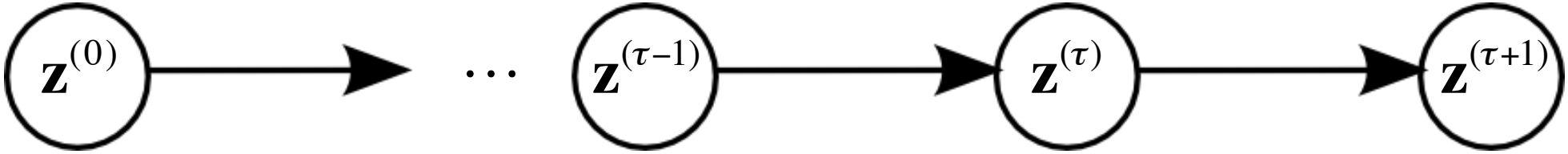
Literature

- Bayesian inference using Metropolis-Hastings: Rogers & Girolami Ch. 4.0 – 4.3, 4.5
- Properties of MCMC methods: Rogers & Girolami Ch. 9.3 – 9.4

- Suggestions for further reading on MCMC:
  - Pierre Brémaud. Markov Chains, Gibbs Fields, Monte Carlo Simulation, and Queues. Springer, 1999.
  - Gerhard Winkler. Image Analysis, Random Fields and Markov Chain Monte Carlo Methods – A Mathematical Introduction. Springer, 2nd edition, 2003.

Extra Slides On Markov Chains

A discrete time Markov chain?



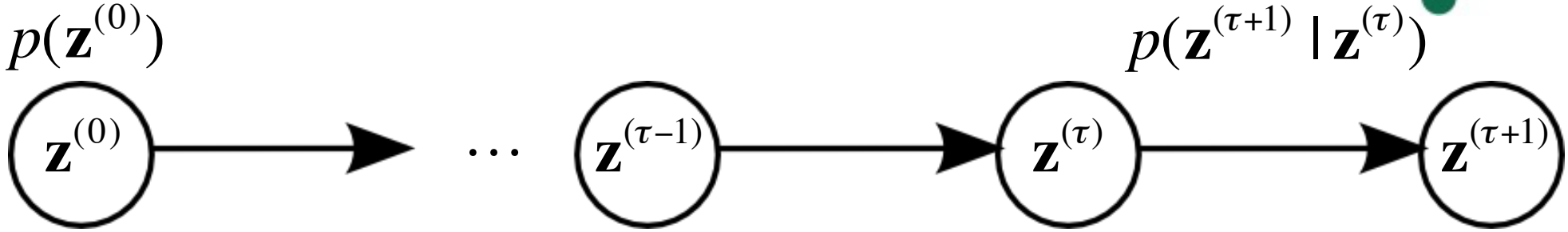
- A discrete time stochastic process is a first order Markov chain (MC), if

$$p(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)}, \mathbf{z}^{(\tau-1)}, \ldots, \mathbf{z}^{(0)}) = p_{\tau}(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)}) \quad \text{(Markov property)}$$

- Its Homogeneous (stationary), if $p_{\tau}(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)}) = p(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)})$
- Its fully specified by $p(\mathbf{z}^{(0)})$ and transition $p(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)})$
- Taking τ steps:

$$p(\mathbf{z}^{(0:\tau)}) = p(\mathbf{z}^{(0)}, \ldots, \mathbf{z}^{(\tau)}) = p(\mathbf{z}^{(0)}) \prod_{n=1}^{\tau} p(\mathbf{z}^{(n)} \mid \mathbf{z}^{(n-1)})$$

A discrete time Markov chain with discrete states



- For discrete state space $\mathbf{z}^{(\tau)} \in \Omega = \{1,2,...,L\}$ for all τ :

  Transition matrix $\mathbf{P} = \{p_{ij}\}_{i,j\in\Omega}$

  $p_{ij} = p(\mathbf{z}^{(\tau+1)} = j \mid \mathbf{z}^{(\tau)} = i)$

  Taking τ steps: $p(\mathbf{z}^{(1:\tau)}) = \mathbf{P}^{\tau} v$

  with initial state vector: $v_i = p(\mathbf{z}^{(0)} = i)$

Example:

$$\begin{matrix} \\ (i=1) \\ (i=2) \end{matrix} \begin{matrix} (j=1 & , & j=2) \\ \begin{bmatrix} 0.2 & & 0.8 \\ 0.9 & & 0.1 \end{bmatrix} \end{matrix}$$

Discrete Markov chains as stochastic state machines – an example

- States: $\Omega = \{a, b, c\}$ and $\mathbf{z}^{(\tau)} \in \Omega$
- Initial state probabilities:

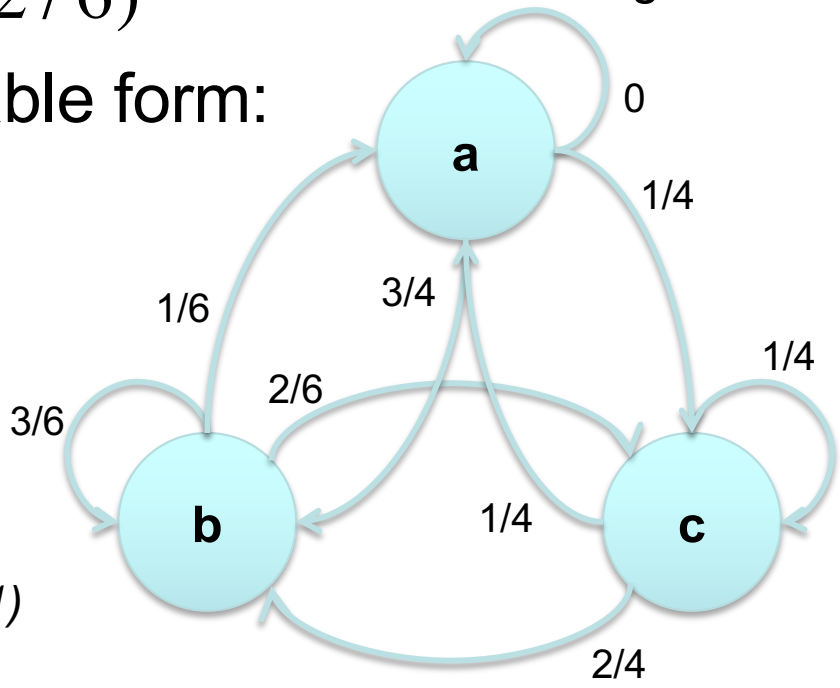
$$p(\mathbf{z}^{(0)} = a) = \frac{1}{6} \quad p(\mathbf{z}^{(0)} = b) = \frac{3}{6} \quad p(\mathbf{z}^{(0)} = c) = \frac{2}{6}$$

- Or in vector form $v = (1/6, 3/6, 2/6)^T$
- Transition probability matrix in table form:

$p(z^{(\tau+1)}\|z^{(\tau)})$	$z^{(\tau+1)}=a$	$z^{(\tau+1)}=b$	$z^{(\tau+1)}=c$
$z^{(\tau)}=a$	0	3/4	1/4
$z^{(\tau)}=b$	1/6	3/6	2/6
$z^{(\tau)}=c$	1/4	2/4	1/4

Rows contain probabilities for $z^{(\tau+1)}$

State diagram

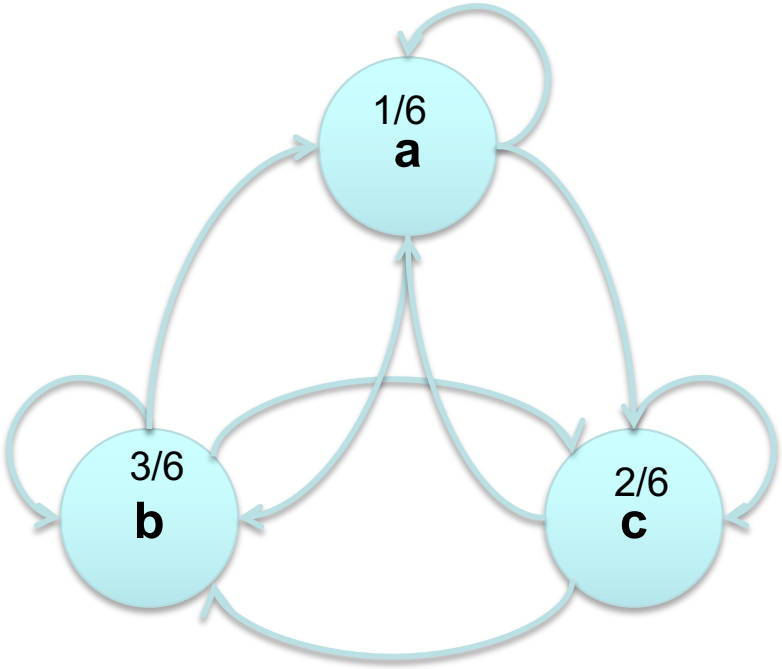


Simulation of a Markov chain – an example

- Sample an initial state from $q(X_0)$

$q(X_0)$	X_0=a	X_0=b	X_0=c
	1/6	3/6	2/6

Result:

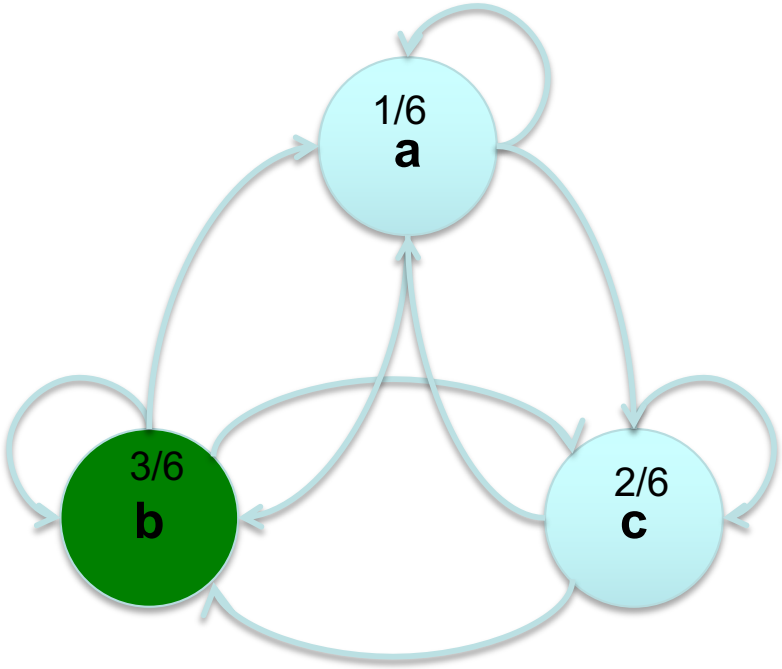


Simulation of a Markov chain – an example

- Sample an initial state from $q(X_0)$

$q(X_0)$	X_0=a	X_0=b	X_0=c
	1/6	3/6	2/6

Result:
b



Simulation of a Markov chain – an example

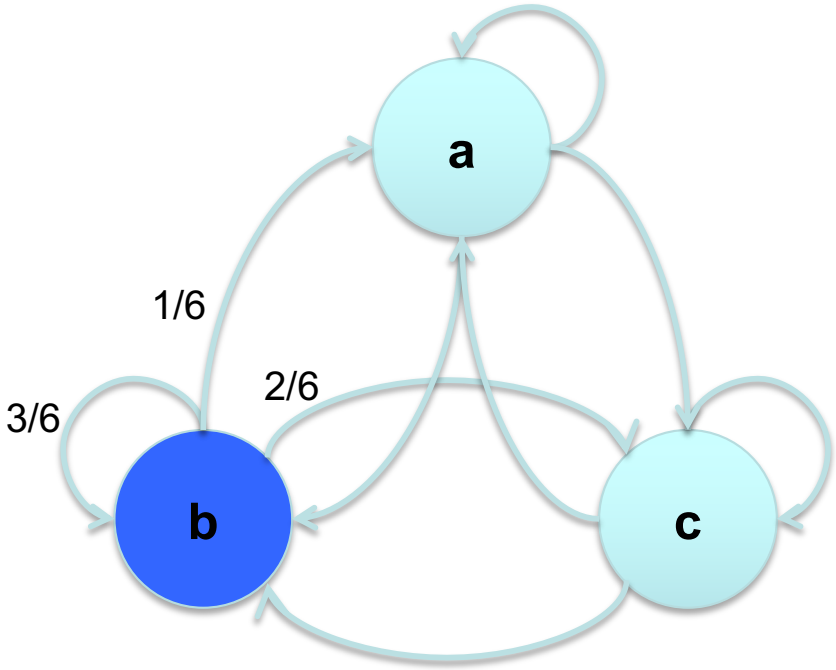
- Sample the next state from the transition probability $p(X_1 \mid X_0 = \mathbf{b})$

$p(X_t \mid X_{t-1})$	X_t=a	X_t=b	X_t=c
X_{t-1}=a	0	3/4	1/4
X_{t-1}=b	1/6	3/6	2/6
X_{t-1}=c	1/4	2/4	1/4

$p(X_1 \mid X_0)$	X_1=a	X_1=b	X_1=c
X_0=b	1/6	3/6	2/6

Result:

b



Simulation of a Markov chain – an example

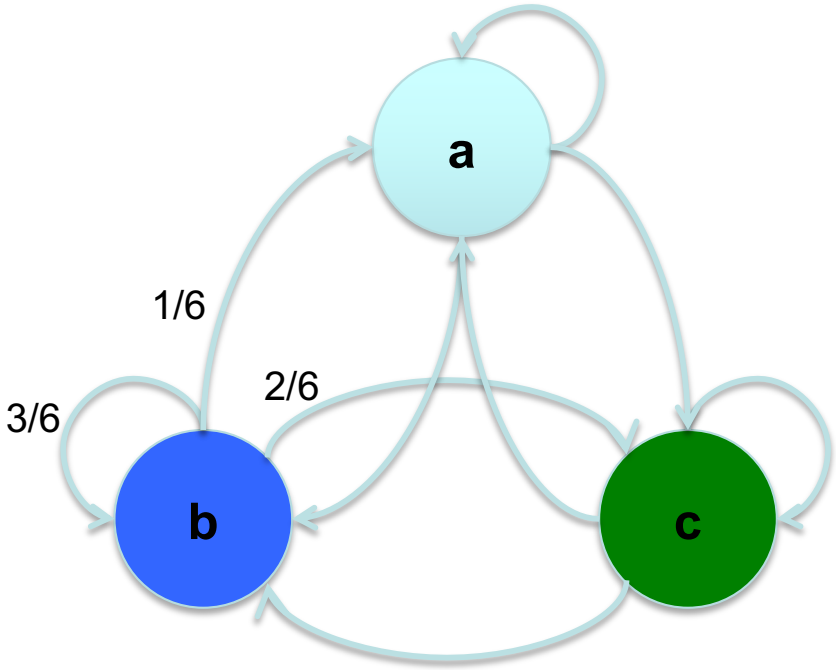
- Sample the next state from the transition probability $p(X_1 \mid X_0 = \mathbf{b})$

$p(X_t \mid X_{t-1})$	X_t=a	X_t=b	X_t=c
X_{t-1}=a	0	3/4	1/4
X_{t-1}=b	1/6	3/6	2/6
X_{t-1}=c	1/4	2/4	1/4

$p(X_1 \mid X_0)$	X_1=a	X_1=b	X_1=c
X_0=b	1/6	3/6	2/6

Result:

bc



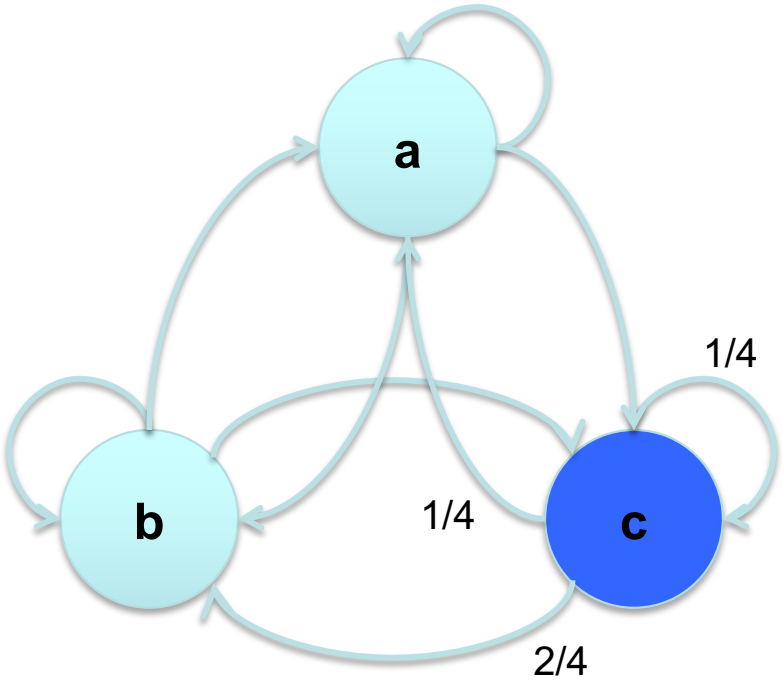
Simulation of a Markov chain – an example

- Sample the next state from the transition probability $p(X_2 \mid X_1 = \mathbf{c})$

$p(X_t \mid X_{t-1})$	X_t=a	X_t=b	X_t=c
X_{t-1}=a	0	3/4	1/4
X_{t-1}=b	1/6	3/6	2/6
X_{t-1}=c	1/4	2/4	1/4

$p(X_2 \mid X_1)$	X_2=a	X_2=b	X_2=c
X_1=c	1/4	2/4	1/4

Result:
bc



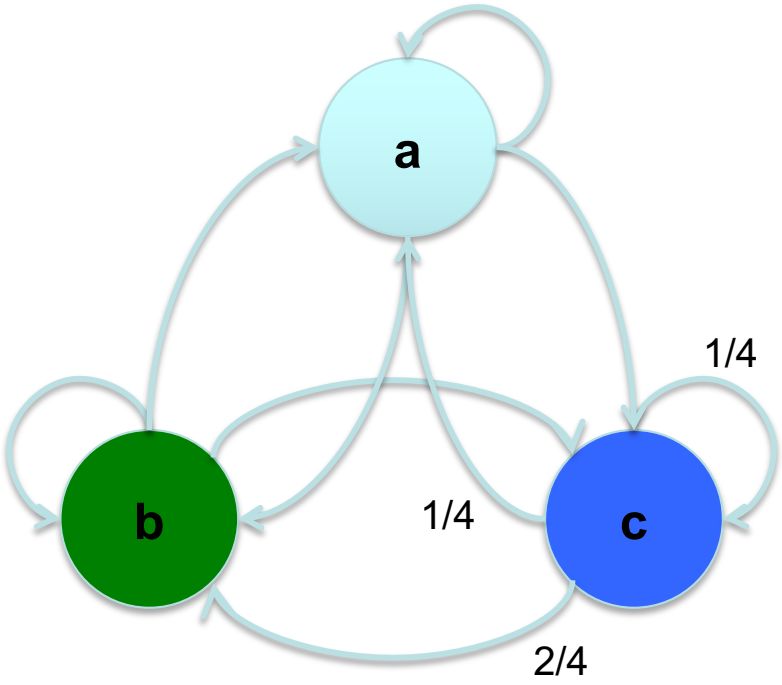
Simulation of a Markov chain – an example

- Sample the next state from the transition probability $p(X_2 \mid X_1 = \mathbf{c})$

$p(X_t \mid X_{t-1})$	X_t=a	X_t=b	X_t=c
X_{t-1}=a	0	3/4	1/4
X_{t-1}=b	1/6	3/6	2/6
X_{t-1}=c	1/4	2/4	1/4

$p(X_2 \mid X_1)$	X_2=a	X_2=b	X_2=c
X_1=c	1/4	2/4	1/4

Result:
bcb



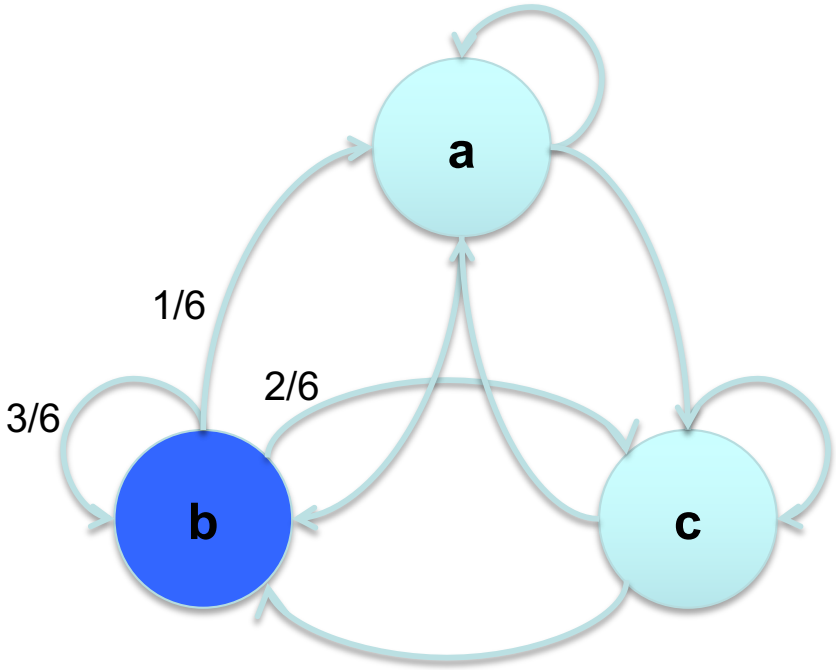
Simulation of a Markov chain – an example

- Sample the next state from the transition probability $p(X_3 \mid X_2 = \mathbf{b})$

$p(X_t \mid X_{t-1})$	X_t=a	X_t=b	X_t=c
X_{t-1}=a	0	3/4	1/4
X_{t-1}=b	1/6	3/6	2/6
X_{t-1}=c	1/4	2/4	1/4

$p(X_3 \mid X_2)$	X_3=a	X_3=b	X_3=c
X_2=b	1/6	3/6	2/6

Result:
bcb



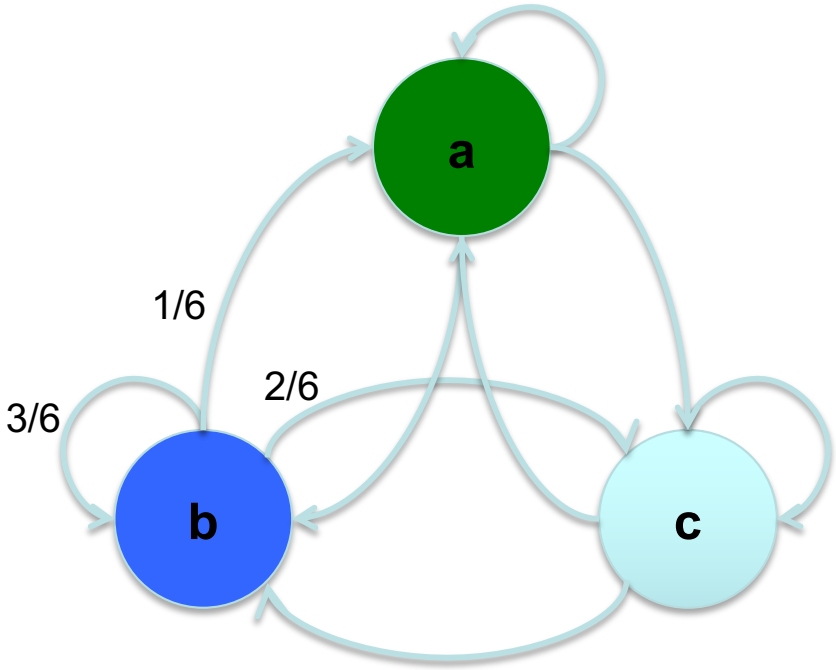
Simulation of a Markov chain – an example

- Sample the next state from the transition probability $p(X_3 \mid X_2 = \mathbf{b})$

$p(X_t \mid X_{t-1})$	X_t=a	X_t=b	X_t=c
X_{t-1}=a	0	3/4	1/4
X_{t-1}=b	1/6	3/6	2/6
X_{t-1}=c	1/4	2/4	1/4

$p(X_3 \mid X_2)$	X_3=a	X_3=b	X_3=c
X_2=b	1/6	3/6	2/6

Result:
bcba



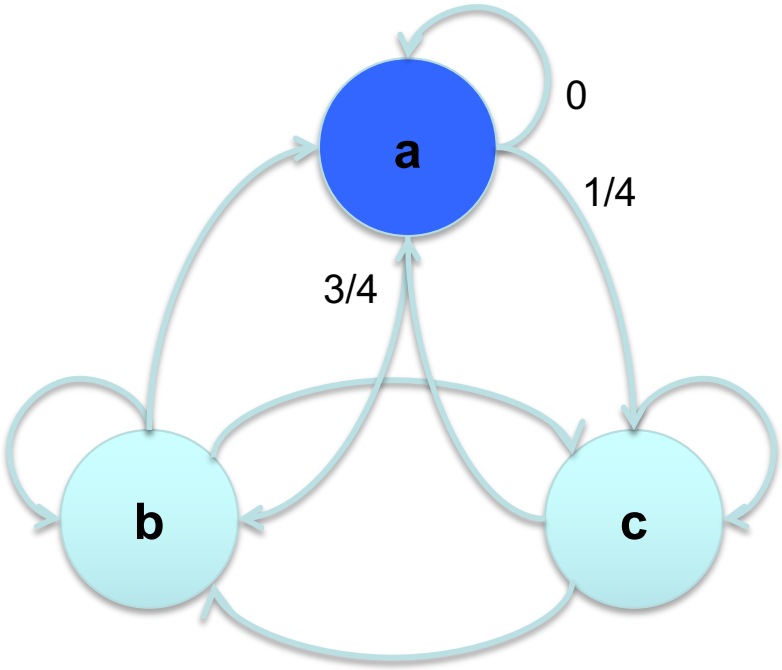
Simulation of a Markov chain – an example

- Sample the next state from the transition probability $p(X_4 \mid X_3 = \mathbf{a})$

$p(X_t \mid X_{t-1})$	X_t=a	X_t=b	X_t=c
X_{t-1}=a	0	3/4	1/4
X_{t-1}=b	1/6	3/6	2/6
X_{t-1}=c	1/4	2/4	1/4

$p(X_4 \mid X_3)$	X_4=a	X_4=b	X_4=c
X_3=a	0	3/4	1/4

Result:
bcba



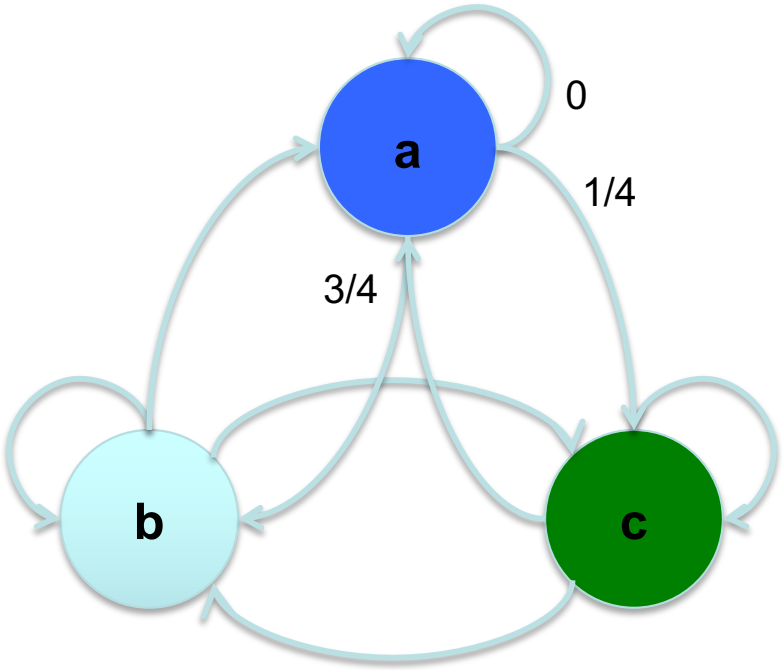
Simulation of a Markov chain – an example

- Sample the next state from the transition probability $p(X_4 \mid X_3 = \mathbf{a})$

$p(X_t \mid X_{t-1})$	X_t=a	X_t=b	X_t=c
X_{t-1}=a	0	3/4	1/4
X_{t-1}=b	1/6	3/6	2/6
X_{t-1}=c	1/4	2/4	1/4

$p(X_4 \mid X_3)$	X_4=a	X_4=b	X_4=c
X_3=a	0	3/4	1/4

Result:
bcbac



Some more properties of Markov chains
What we need for MCMC to work

- We want a Markov chain such that $p(\mathbf{z})$ is an invariant.
- We also require that $p(\mathbf{z}^{(\tau)})$ converge to $p(\mathbf{z})$ as $\tau \rightarrow \infty$, irrespectively of the choice of initial distribution $q(\mathbf{z}^{(0)})$.
- Some initial burn-in time is to be expected before convergence is achieved.

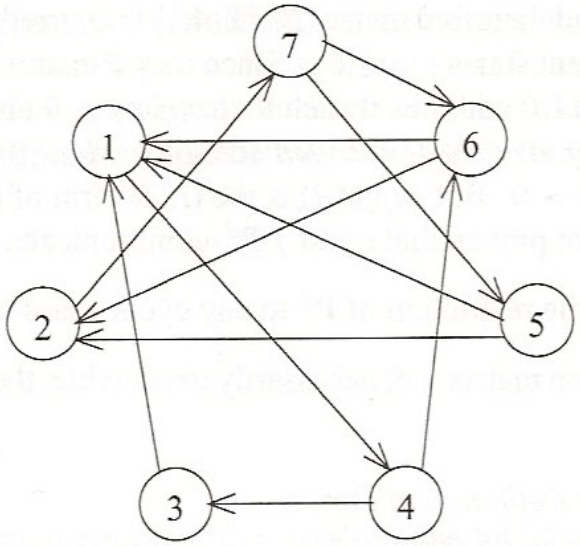
- Such a Markov chain is called an *ergodic* Markov chain.

- A sufficient condition for ergodicity is that the conditional distributions $p(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)})$ are nowhere zero, $p(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)}) > 0$ for all $\mathbf{z}^{(\tau+1)}, \mathbf{z}^{(\tau)}$.
- This ensures that any state can be reached from any other state in a finite number of steps.

Ergodicity: $p(\mathbf{z}^{(\tau+1)} \mid \mathbf{z}^{(\tau)}) > 0$, a sufficient condition, but not a necessary condition

- Here is an example of a state transition diagram which leads to an ergodic Markov chain:
- With cyclic state classes:

$$C_0 = \{1,2\} \;,\; C_1 = \{4,7\} \;,\; C_2 = \{3,5,6\}$$



- And transition probability matrix:

$$\mathbf{P} = \begin{array}{c} \\ C_0 \\ C_1 \\ C_2 \end{array} \begin{array}{c} \begin{array}{ccc} C_0 & C_1 & C_2 \end{array} \\ \begin{bmatrix} 0 & A_0 & 0 \\ 0 & 0 & A_1 \\ A_2 & 0 & 0 \end{bmatrix} \end{array}$$