Markov Chains
and
Markov Chain Monte Carlo

Yee Whye Teh
Department of Statistics
http://www.stats.ox.ac.uk/~teh
TAs: Luke Kelly, Lloyd Elliott

Practical

Practicals

- Some mathematical derivations.
- Some programming in:
  - R
  - MATLAB
- Probably not possible to do all practicals; pick and choose.

- Package available at

  http://www.stats.ox.ac.uk/~teh/teaching/dtc2014

Schedule

- 0930-1100 Lecture: Introduction to Markov chains
- 1100-1200 Practical
- 1200-1300 Lecture: Further Properties of Markov chains
- 1300-1400 Lunch
- 1400-1515 Practical
- 1515-1630 Practical *change*
- 1630-1730 Lecture: Continuous-time Markov chains

- 0930-1100 Lecture: Introduction to Markov chain Monte Carlo methods
- 1100-1230 Practical
- 1230-1330 Lunch
- 1330-1500 Lecture: Further Markov chain Monte Carlo methods
- 1500-1700 Practical
- 1700-1730 Wrap-up

Markov Chains

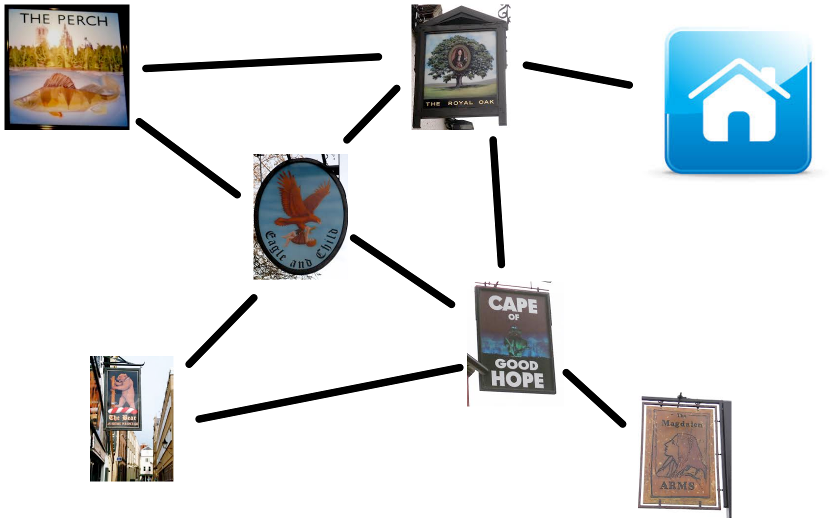
Andrey Andreyevich Markov
1856-1922

Sequential Processes

- Sequence of random variables X_0, X_1, X_2, X_3,...
- Not iid (independently and identically distributed).
- Examples:
  - X_i = Rain or shine on day i.
  - X_i = Nucleotide base at position i.
  - X_i = State of system at time i.
- Joint probability can be factorized using Bayes' Theorem:

$$\begin{aligned}&\mathbb{P}(X_0 = x_0, X_1 = x_1, X_2 = x_2 \ldots)\\ =&\mathbb{P}(X_0 = x_0)\mathbb{P}(X_1 = x_1 | X_0 = x_0)\mathbb{P}(X_2 = x_2 | X_0 = x_0, X_1 = x_1)\cdots\end{aligned}$$

Random Pub Crawl

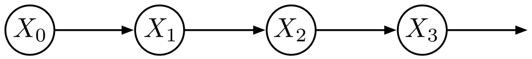


Markov Assumption

- Markov Assumption: each X_i **only** depends on the previous X_{i-1}.

$$\begin{aligned}&\mathbb{P}(X_0 = x_0, X_1 = x_1, X_2 = x_2 \ldots)\\ =&\mathbb{P}(X_0 = x_0)\mathbb{P}(X_1 = x_1 | X_0 = x_0)\mathbb{P}(X_2 = x_2 | X_0 = x_0, X_1 = x_1)\cdots\\ =&\mathbb{P}(X_0 = x_0)\mathbb{P}(X_1 = x_1 | X_0 = x_0)\mathbb{P}(X_2 = x_2 | X_1 = x_1)\cdots\end{aligned}$$

- Future is independent of the past, given the present.
- Process "has no memory".



- Higher order Markov chains:

$$\begin{aligned}&\mathbb{P}(X_t = x_t | X_0 = x_0, \ldots, X_{t-1} = x_{t-1})\\ =&\mathbb{P}(X_t = x_t | X_{t-k} = x_{t-k}, \ldots, X_{t-1} = x_{t-1})\end{aligned}$$

Jukes-Cantor DNA Evolution

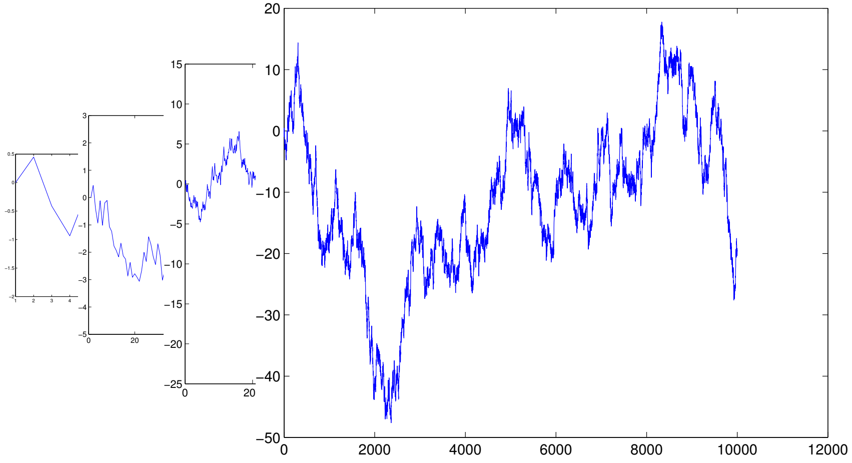
GCTCATGCCG
|
GCTAATGCCG
|
GCTATTGGCG

	$\to A$	$\to G$	$\to C$	$\to T$
A	$1-3\epsilon$	ϵ	ϵ	ϵ
G	ϵ	$1-3\epsilon$	ϵ	ϵ
C	ϵ	ϵ	$1-3\epsilon$	ϵ
T	ϵ	ϵ	ϵ	$1-3\epsilon$

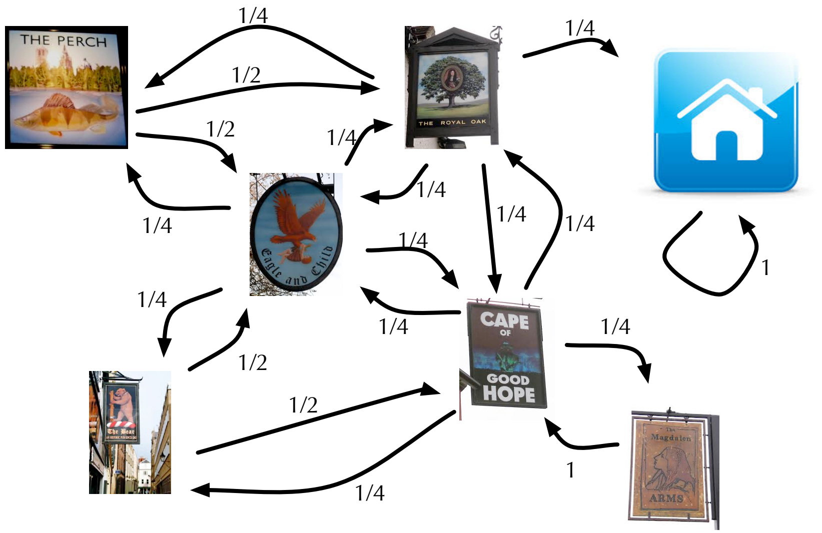
- Mutation process operates independently at each position.
- Small total probability 3ϵ of a mutation happening at each generation.

Random Walk on $\mathbb{Z}$

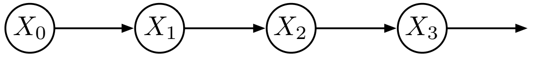
- Start at 0.
- Move up or down with probability 1/2.



State Transition Diagrams



Parameterization



- Initial distribution:

$$\mathbb{P}(X_0 = i) = \lambda_i$$

- Transition probability matrix:

$$\mathbb{P}(X_t = j | X_{t-1} = i) = T_{ij}$$

  - Homogeneous Markov chains (transition probabilities do not depend on the time step)
  - Inhomogeneous Markov chains - transitions do depend on time step.

Practical

Simulating Random Pub Crawl (*)

- Write a programme to simulate from the random pub crawl. (From the “home” state allow probability 1/2 of going back to the Royal Oak).
- Starting from the Home state, run your programme 1000 times, each time simulating a Markov chain of length 100.
- Each simulation should be a random sequence of values (s_1,s_2,s_3...,s_{100}) where each s_i is a pub.
- Collect statistics of the number of times each state is visited at each time step *t* = 1...100.
- How do the statistics defer if you started at Magdelen Arms?
- Does the distribution over states visited at step t converge for large t?
- Approximately how long does it take for the chain to “forget” whether it started at Home or at Magdelen Arms?

Useful Properties of Markov Chains

Marginal Distributions

- Similarly we can calculate the marginal probabilities of each X_i recursively:

$$\begin{aligned} P_i^{(t)} :=& \mathbb{P}(X_t = i) \\ =& \sum_k \mathbb{P}(X_{t-1} = k)\mathbb{P}(X_t = i | X_{t-1} = k) \\ =& \sum_k P_k^{(t-1)} T_{ki} \\ =& (\lambda T^t)_i \end{aligned}$$

- where we take λ to be a row vector.

Chapman-Kolmogorov Equations

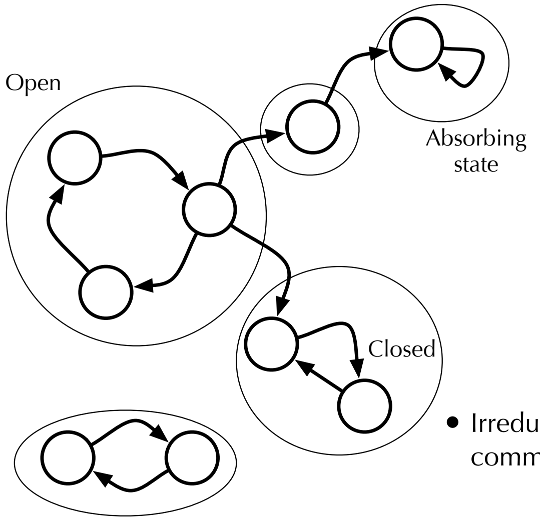
- We can calculate multi-step transition probabilities recursively:

$$\begin{aligned} \mathbb{P}(X_{t+2} = j | X_t = i) =& \sum_k \mathbb{P}(X_{t+2} = j | X_{t+1} = k)\mathbb{P}(X_{t+1} = k | X_t = i) \\ =& \sum_k T_{ik} T_{kj} \\ =& (T^2)_{ij} \end{aligned}$$

- Similarly:

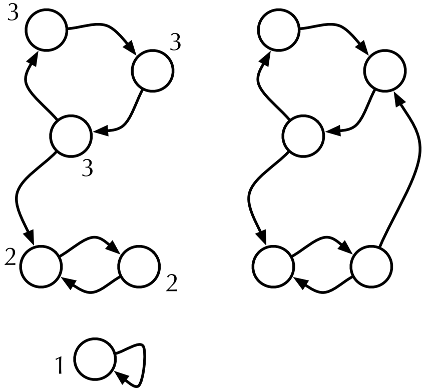
$$\begin{aligned} P_{ij}^{(m)} :=& \mathbb{P}(X_{t+m} = j | X_t = i) \\ =& \sum_k \mathbb{P}(X_{t+m} = j | X_{t+1} = k)\mathbb{P}(X_{t+1} = k | X_t = i) \\ =& \sum_k P_{ik}^{(m-1)} T_{kj} \\ =& (T^m)_{ij} \end{aligned}$$

Communicating Classes



- Irreducible: One communicating class.

Periodicity

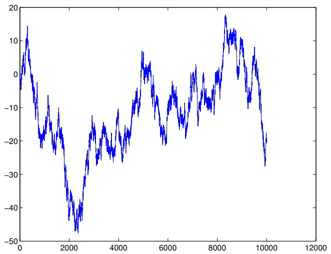


- Period of i:

$$gcd\{n : \mathbb{P}(\text{returning to } i \text{ from } i \text{ in } n \text{ steps}) > 0\}$$

- If a chain is irreducible, then all states have the same period.
- If the period is 1, then we say the chain is aperiodic.

Random Walk on $\mathbb{Z}$



- Start at 0.
- Move up or down with probability 1/2.
- X_{2t} is the sum of 2t iid {+1,-1} variables.
- It equals 0 if there are exactly t +1's, and t -1's.
- This probability is:

$$P_{00}^{(2t)} = \frac{(2t)!}{t!t!}\left(\frac{1}{2}\right)^{2t} \approx \frac{1}{\sqrt{\pi}\sqrt{t}}$$

- Using Stirling's Formula:

$$n! \approx \sqrt{2\pi} n^{n+1/2} e^{-n}$$

- This sums to infinity over *t*, so chain is recurrent.

Recurrence and Transience

- If we start at state *i*, what is the chance that we will return to *i*?
- Two possibilities:

$$\mathbb{P}(\exists t > 0 : X_t = i | X_0 = i) = p < 1$$

  - Total number of times we will encounter *i* in all future will be finite.
  - State *i* is transient.

$$\mathbb{P}(\exists t > 0 : X_t = i | X_0 = i) = 1$$

  - We will return to *i* infinitely often.
  - State *i* is recurrent.
- A state *i* is recurrent if and only if

$$\sum_t P_{ii}^{(t)} = \infty$$

Positive Recurrence and Null Recurrence

- Recurrence:
  - Chain will revisit a state infinitely often.
  - From state *i* we will return to *i* after a (random) finite number of steps.
- But the expected number of steps can be infinite!
  - This is called null recurrence.
- If expected number of steps is finite, this is called positive recurrent.
- Example: random walk on $\mathbb{Z}$.

Practical

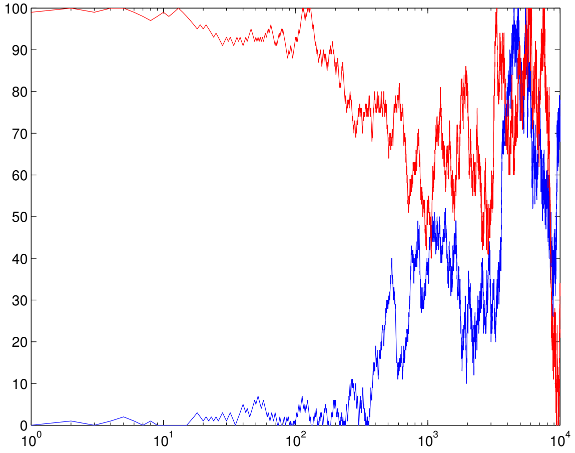
Communicating Classes

- Find the communicating classes and determine whether each class is open or closed, and the periodicity of the closed classes.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad \begin{pmatrix} 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 1/4 & 0 & 3/4 & 0 \\ 0 & 0 & 1/3 & 0 & 2/3 \\ 1/4 & 1/2 & 0 & 1/4 & 0 \\ 1/3 & 0 & 1/3 & 0 & 1/3 \end{pmatrix}$$

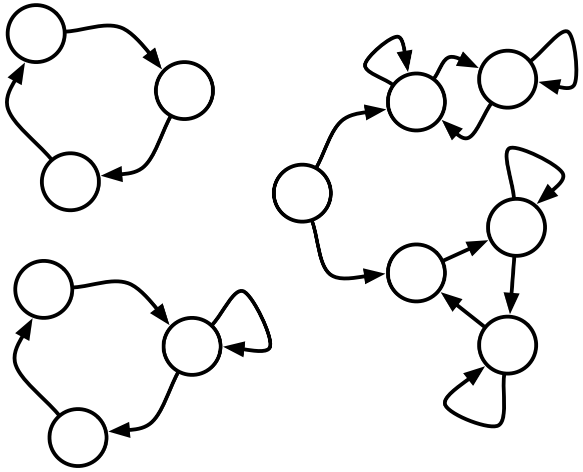
Do Markov Chains Forget?

- A Markov chain on {0,1,2,...,100}.
- At each step: move up or down by 1 at random, except at boundaries.
- Start at 0 and at 100.



Convergence of Markov Chains

Do Markov Chains Forget?



Stationary Distribution

- If a Markov chain “forgets” then for any two initial distributions/ probability vectors λ and γ,

$$\lambda T^n \approx \gamma T^n \quad \text{for large } n$$

- In particular, there is a distribution/probability vector π such that

$$\lambda T^n \to \pi \quad \text{as } n \to \infty$$

- Taking λ = π, we see that

$$\pi T = \pi$$

- Such a distribution is called a stationary or equilibrium distribution.
  - When do Markov chains have stationary distributions?
  - When are stationary distributions unique?

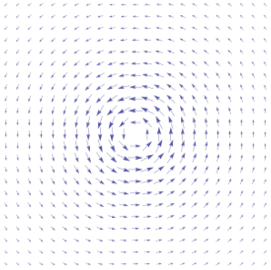
Stationarity and Reversibility

- Global balance: at a stationary distribution, the flow of probability mass into and out of each state has to be balanced:

$$\sum_{i=1}^{K} \pi_i T_{ij} = \pi_j = \sum_{k=1}^{K} \pi_j T_{jk}$$

- Detailed balance: the flow of probability mass between each pair of states is balanced:

$$\pi_i T_{ij} = \pi_j T_{ji}$$



- A Markov chain satisfying detailed balance is called reversible. Reversing the dynamics leads to the same chain.
- Detailed balance can be used to check that a distribution is the stationary distribution of a irreducible, periodic, reversible Markov chain.

Convergence Theorems

- A positive recurrent Markov chain *T* has a stationary distribution.
- If *T* is irreducible and has a stationary distribution, then it is unique and

$$\pi_i = \frac{1}{m_i}$$

  where m_i is the mean return time of state *i*.

- If *T* is irreducible, aperiodic and has stationary distribution π then

$$\mathbb{P}(X_n = i) \to \pi_i \quad \text{as } n \to \infty$$

- (Ergodic Theorem): If *T* is irreducible with stationary distribution π then

$$\frac{\#\{t \leq n : X_t = i\}}{n} \to \pi_i \quad \text{as } n \to \infty$$

Eigenvalue Decomposition

- The stationary distribution is a left eigenvector of T, with eigenvalue 1.

$$\pi T = \pi$$

- All eigenvalues of T have length ≤ 1. (Some eigenvalues can be complex valued).
- Let the eigenvalues be a_1=1, a_2...,a_K, with corresponding left eigenvectors v_1=π, v_2,...,v_K. Then:

$$\lambda = b_1\pi + \sum_{k=2}^{K} b_k v_k$$

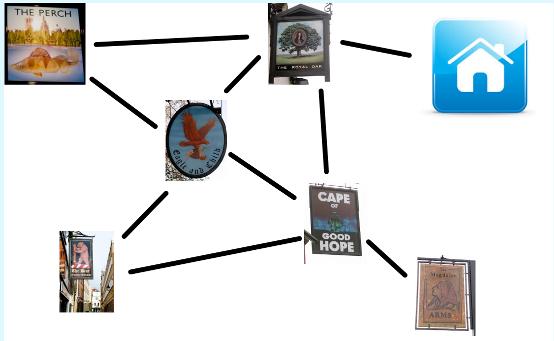
$$\lambda T^n = b_1\pi + \sum_{k=2}^{K} b_k a_k^n v_k \to b_1\pi \quad \text{as } n \to \infty, \text{ if all } |a_k| < 1 \text{ for } k \geq 2$$

- (Bit of algebra shows that b_1=1.)
- If there is another eigenvector with eigenvalue 1, then stationary distribution is not unique.

Practical

Random Walk

- Show that a random walk on a connected graph is reversible, and has stationary distribution π with π_i proportional to deg(i), the number of edges connected to i.
- What is the probability the drinker is at home at Monday 9am if he started the pub crawl on Friday?



Additional Points

- Transition probability matrix T has to have:
  - non-negative entries
  - rows that sum to 1
- Any such matrix is a transition probability matrix.

- Periodicity demonstration.

Practical

Stationary Distributions

- Solve for the (possibly not unique) stationary distribution(s) of the following Markov chains.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad \begin{pmatrix} 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 1/4 & 0 & 3/4 & 0 \\ 0 & 0 & 1/3 & 0 & 2/3 \\ 1/4 & 1/2 & 0 & 1/4 & 0 \\ 1/3 & 0 & 1/3 & 0 & 1/3 \end{pmatrix}$$

Estimating Markov Chains

Maximum Likelihood Estimation

- Observe a sequence *x*$_0$*, x*$_1$*, x*$_2$*, x*$_3$*,… x*$_t$.
- Likelihood of the sequence under the Markov chain model is:

$$\mathcal{L}(\lambda, T) = \lambda_{x_0} \prod_{s=1}^{t} T_{x_{s-1}x_s} = \lambda_{x_0} \prod_{i=1}^{K} \prod_{j=1}^{K} T_{ij}^{N_{ij}}$$

  where N_{ij} is the number of observed transitions *i* -> *j*.
- We can solve for the maximum likelihood estimator:

$$T_{ij} = \frac{N_{ij}}{\sum_{k=1}^{K} N_{ik}}$$

Practical

Further Estimation Procedures

- Bayesian inference for the transition matrix:
  - Show that you can construct a conjugate prior for *T* using Dirichlet distributions.
  - What is the corresponding posterior given observe sequence?
  - What is the marginal probability of the data?
  - What is the posterior mean of *T* and how does it relate to the ML estimator?
- Derive the maximum likelihood estimator for a Markov chain of order *h*.

Practical

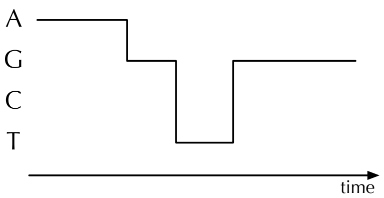
Markov Model of English Text (*)

- Download a large piece of English text, say “War and Peace” from Project Gutenberg.
- We will model the text as a sequence of characters.
- Write a programme to compute the ML estimate for the transition probability matrix.
- You can use the file markov_text.R or markov_text.m to help convert from text to the sequence of states needed. There are *K* = 96 states and the two functions are text2states and states2text.
- Generate a string of length 200 using your ML estimate.
- Does it look sensible?

Continuous-Time Markov Chains

Jukes-Cantor DNA Evolution

	$\to A$	$\to G$	$\to C$	$\to T$
A	$1-3\epsilon$	ϵ	ϵ	ϵ
G	ϵ	$1-3\epsilon$	ϵ	ϵ
C	ϵ	ϵ	$1-3\epsilon$	ϵ
T	ϵ	ϵ	ϵ	$1-3\epsilon$



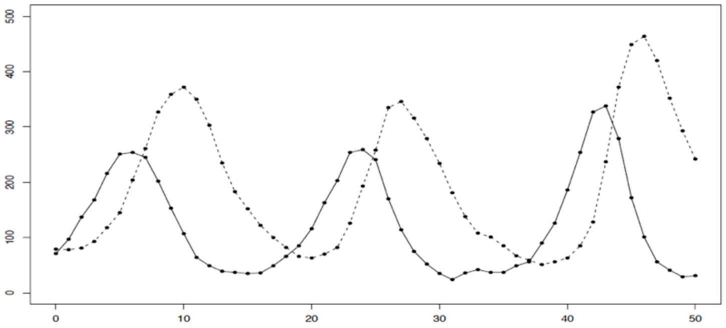
- Probability of mutation is O(ε) per generation.
- mutations will appear at rate of once every O(1/ε) generations.
- Measuring time in units of 1/ε leads to a continuous-time Markov chain.
- In each time step of length ε, total probability of a mutation is 3ε.

$$P = \begin{pmatrix} 1-3\epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 1-3\epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 1-3\epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 1-3\epsilon \end{pmatrix} = I + \epsilon \begin{pmatrix} -3 & 1 & 1 & 1 \\ 1 & -3 & 1 & 1 \\ 1 & 1 & -3 & 1 \\ 1 & 1 & 1 & -3 \end{pmatrix}$$

Lotka-Volterra Process (Predator-Prey)

- A continuous-time Markov chain over $\mathbb{N}^2$, number of predators and preys in an ecosystem.

$$R(\{x,y\} \to \{x+1,y\}) = \alpha x \qquad R(\{x,y\} \to \{x-1,y\}) = \beta xy$$
$$R(\{x,y\} \to \{x,y+1\}) = \delta xy \qquad R(\{x,y\} \to \{x,y-1\}) = \gamma y$$



Continuous-time Markov Chains

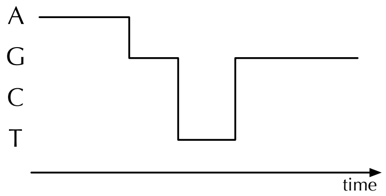
- A collection of random variables $(X_t)_{t\geq 0}$.
- An initial distribution λ and a transition rate matrix *R*.
- Suppose $X_t = i$. Then in the next ε time,

$$\mathbb{P}(X_{t+\epsilon} = j | X_t = i) = I_{ij} + \epsilon R_{ij}$$

$$\begin{pmatrix} -3 & 1 & 1 & 1 \\ 1 & -3 & 1 & 1 \\ 1 & 1 & -3 & 1 \\ 1 & 1 & 1 & -3 \end{pmatrix}$$

  - Rows of R sum to 0.
  - Off-diagonal entries are non-negative.
  - On-diagonal entries are negative of sum of off-diagonal ones.

Gillespie's Algorithm



- Start by sampling X_0 from initial distribution λ.
- When in state i, wait in state for an amount of time distributed as

$$\mathrm{Exp}(|R_{ii}|)$$

- At end of waiting time, transition to a different state j ≠ i with probability

$$\frac{(R_{i1}, \ldots, R_{ij-1}, 0, R_{ij+1}, \ldots, R_{iK})}{|R_{ii}|}$$

Chapman-Kolmogorov Equations

- Denote *P*(*t*) as the transition probability matrix over time interval *t*.
- Transition probabilities can be computed using matrix exponentiation:

$$\begin{aligned} P(t)_{ij} :=& \mathbb{P}(X_t = j | X_0 = i) \\ =& \mathbb{P}(X_{tn\frac{1}{n}} = j | X_0 = i) \\ \approx& ((I + \tfrac{1}{n}R)^{tn})_{ij} \to \exp(tR)_{ij} \end{aligned}$$

- Composition of transition probability matrices:

$$P(t+s) = \exp((t+s)R) = \exp(tR)\exp(sR) = P(t)P(s)$$

- Forward/backward equations:

$$\begin{aligned} P(t+\epsilon) &= P(t)(I+\epsilon R) \\ \frac{\partial P(t)}{\partial t} &\approx \frac{P(t+\epsilon)-P(t)}{\epsilon} \to P(t)R = RP(t) \end{aligned}$$

Reversibility and Detailed Balance

- If a Markov chain with rate matrix *R* has reached its stationary distribution π, then flow of probability mass into and out of states is balanced.
- Global balance:

$$\begin{aligned} \sum_{i=1}^{K} \pi_i R_{ij} = && 0 && = \sum_{k=1}^{K} \pi_j R_{jk} \\ \sum_{i \neq j} \pi_i R_{ij} = && \pi_j |R_{jj}| && = \sum_{k \neq j} \pi_j R_{jk} \end{aligned}$$

- Detailed balance for reversible chains:

$$\pi_i R_{ij} = \pi_j R_{ji}$$

Convergence to Stationary Distribution

- Suppose we have an
  - irreducible,
  - aperiodic and
  - positive recurrent

  continuous-time Markov chain with rate matrix *R*.
- Then it has a unique stationary distribution π which it converges to:

$$\begin{aligned} \mathbb{P}(X_t = i) \to \pi_i \qquad & \text{as } t \to \infty \\ \frac{1}{T}\int_0^T \mathbf{1}(X_t = i) \to \pi_i \qquad & \text{as } t \to \infty \end{aligned}$$

Kimura 80 Model

- Rate matrix:

	$\to A$	$\to G$	$\to C$	$\to T$
A	$-\kappa - 2$	κ	1	1
G	κ	$-\kappa - 2$	1	1
C	1	1	$-\kappa - 2$	κ
T	1	1	κ	$-\kappa - 2$

- Distinguish between transitions A↔G (purine) and C↔T (pyrimidine) and transversions.

- Practical: show that the stationary distribution of K80 model is uniform over {A,G,C,T}.

Felsenstein 81 Model

- Rate matrix:

$$\begin{array}{c|cccc} & \to A & \to G & \to C & \to T \\ \hline A & -\pi_G - \pi_C - \pi_T & \pi_G & \pi_C & \pi_T \\ G & \pi_A & -\pi_A - \pi_C - \pi_T & \pi_C & \pi_T \\ C & \pi_A & \pi_G & -\pi_A - \pi_G - \pi_T & \pi_T \\ T & \pi_A & \pi_G & \pi_C & -\pi_A - \pi_G \pi_C \end{array}$$

- Incoming rates to each state are all the same.

- Practical: Find the stationary distribution of the F81 model.

Monte Carlo Methods

Monte Carlo

Practical

Predator-Prey Model (*)

- Use Gillespie's algorithm to simulate from the predator-prey model:

$$R(\{x,y\} \to \{x+1,y\}) = \alpha x \qquad R(\{x,y\} \to \{x-1,y\}) = \beta xy$$
$$R(\{x,y\} \to \{x,y+1\}) = \delta xy \qquad R(\{x,y\} \to \{x,y-1\}) = \gamma y$$

- You can represent the continuous-time trajectory using a sequence of pairs (t_0,s_0), (t_1,s_1),... (t_A,s_A), where
  - $t_0 = 0$, s_0 is the initial state at time 0,
  - each subsequent pair (t_a,s_a) is the next state and the time the chain jumps to the state.
- How is the dynamics of the model affected by the parameters α, β, γ, δ?

Bayesian Inference

- A model described as a joint distribution over a collection of variables:
  - X - collection of variables, with observed value x.
  - Y - collection of variables which we would like to learn about.
- Assume it has density $p(x,y)$.
- Two quantities of interest
  - The posterior distribution of Y given $X = x$:

$$p(y|x) = \frac{p(x,y)}{p(x)} \qquad\qquad \int f(y)p(y|x)dy$$

  - The marginal probability of observation x:

$$p(x) = \int p(x,y)dy$$

Decision Theory

- Given observation *x*, we would like to make a decision:
  - Decide on an optimal action to take
  - Decide on a prediction to make
- Loss function *L*(*a*,(*x*,*y*)). The decision minimizing expected loss is:

$$\arg\min_a \int L(a,(x,y))p(y|x)dy$$

- Example:
  - Y = whether a patient has disease.
  - X = status of medical test.
  - a = whether doctor diagnoses disease.

$$L(a,(x,y)) = \begin{cases} 0 & \text{if } a = y, \\ 1 & \text{if } a = T,\ y = F, \\ 100 & \text{if } a = F,\ y = T. \end{cases}$$

The Monte Carlo Method

- Interested in evaluating the expectation of a test function:

$$\theta = \mathbb{E}_{p(y|x)}[f(y)] = \int f(y)p(y|x)dy$$

- Analytic integration: limited applicability and limited model realism.
- Numerical integration: intractable.
- Strong law of large numbers:
  - Draw iid samples $y_n \sim p(y|x)$.

$$\theta \approx \frac{1}{N}\sum_{n=1}^{N} f(y_n)$$

- Unbiased. Variance *O*(1/*N*).
- Central limit theorem can characterize deviations away from θ.

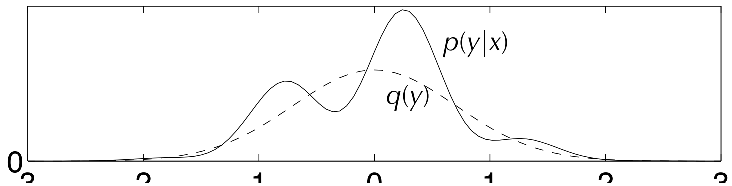
Importance Sampling

- Often not trivial or impossible to sample from *p*(*y*|*x*).
- Use a tractable proposal distribution *q*(*y*) instead.
  - Draw iid samples $y_n \sim q(y)$.

$$\int f(y)p(y|x)dy = \int f(y)\frac{p(y|x)}{q(y)}q(y)dy \approx \frac{1}{N}\sum_{n=1}^{N} f(y_n)\frac{p(y_n|x)}{q(y_n)}$$

- A weighted average with weights

$$w(y_n) = \frac{p(y_n|x)}{q(y_n)}$$



Importance Sampling

- Unbiased.

$$\int f(y)p(y|x)dy = \int f(y)\frac{p(y|x)}{q(y)}q(y)dy \approx \frac{1}{N}\sum_{n=1}^{N} f(y_n)\frac{p(y_n|x)}{q(y_n)}$$

- Variance can be smaller or larger.

$$\frac{1}{N}\mathbb{V}[f(y)w(y)] = \frac{1}{N}\left(\mathbb{E}[f(y)^2w(y)^2] - \mathbb{E}[f(y)w(y)]^2\right)$$

- Important for *q*(*y*) to be large whenever *p*(*y*|*x*) is large.
- Effective sample size can be estimated using

$$1 \leq \frac{\left(\sum_{n=1}^{N} w(y_n)\right)^2}{\sum_{n=1}^{N} w(y_n)^2} \leq N$$

Importance Sampling

- Often we can only evaluate *p*(*y*|*x*) up to normalization constant:

$$p(y|x) = \frac{\tilde{p}(y,x)}{Z(x)}$$

- where $\tilde{p}(y,x)$ can be computed but not *Z*(*x*).
- In these situations we can estimate *Z*(*x*) and θ as follows:

$$Z(x) = \int \tilde{p}(y,x)dy = \int \frac{\tilde{p}(y,x)}{q}(y)q(y)dy \approx \frac{1}{N}\sum_{n=1}^{N} w(y_n), \quad w(y) = \frac{\tilde{p}(y,x)}{q(y)}$$

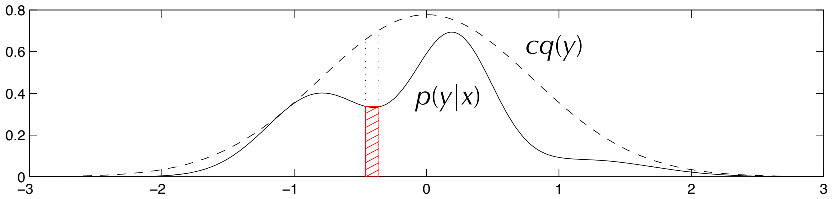
$$\theta = \frac{\int f(y)\tilde{p}(y,x)dy}{\int \tilde{p}(y,x)dy} \approx \frac{\sum_{n=1}^{N} f(y_n)w(y_n)}{\sum_{n=1}^{N} w(y_n)}$$

Rejection Sampling

- Find a proposal distribution *q*(*y*) and a constant c>0 which upper bounds *p*(*y*|*x*):

$$p(y|x) \leq cq(y) \quad \text{for all } y$$

- Repeat { sample *y** ~ *q*(*y*), sample *u* ~ Uniform[0, *cq*(*y**)] } until *u*<*p*(*y**).
- Return *y** as an exact sample from *p*(*y*|*x*).
- Unbiased.
- Expected number of samples is 1/*c*.



Markov Chain Monte Carlo

Nicolas Metropolis
1915-1999

The Monte Carlo Method

- Strong law of large numbers:
  - Draw iid samples y_n ~ *p*(*y*|*x*).

$$\theta \approx \frac{1}{N}\sum_{n=1}^{N} f(y_n)$$

- Ergodic theorem:
  - Construct irreducible, aperiodic, positive recurrent Markov chain with stationary distribution *p*(*y*|*x*).
  - Simulate y_1, y_2, y_3,... from markov chain. Then:

$$\frac{1}{N}\sum_{n=1}^{N} f(y_n) \rightarrow \theta \quad \text{as } N \rightarrow \infty$$

- Never as good as iid samples, but much wider applicability.

Success Stories

- Building the first nuclear bomb.
- Estimating orbits of exoplanets.
- Automated image analysis and edge detection.
- Computer game playing.
- Running web searches.
- Calculation of 3D protein folding structure.
- Determine population structure from genetic data.
- etc etc etc
- Metropolis and Ulam 1949. The Monte Carlo method. Journal of the American Statistical Association 44:335-341.
- Gelfand and Smith 1990. Sampling-based approaches to calculating marginal densities. Journal of the American Statistical Association 85:398-409.

Metropolis Algorithm

- Use detailed balance condition to verify that stationary distribution of constructed Markov chain is $\pi(y)$.
- Suppose y and y' are two distinct states.
- By symmetry we can assume $\pi(y) < \pi(y')$.
- Starting from y, probability of going to y′ is:

$$\pi(y)q(y'|y)\min\left(1, \frac{\tilde{\pi}(y')}{\tilde{\pi}(y)}\right) = \pi(y)q(y'|y)$$

- Starting from y′, probability of going to y is:

$$\pi(y')q(y|y')\min\left(1, \frac{\tilde{\pi}(y)}{\tilde{\pi}(y')}\right) = \pi(y)q(y'|y)$$

Metropolis Algorithm

- We wish to sample from some distribution with density

$$\pi(y) = \frac{\tilde{\pi}(y)}{Z}$$

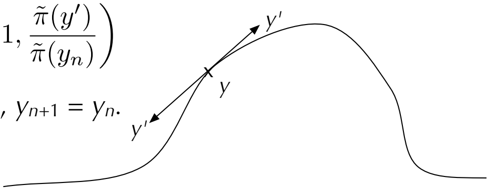
- Suppose the current state is y_n.
  - Propose next state y' from a symmetric proposal distribution $q(y'|y_n)$.

$$q(y'|y) = q(y|y') \quad \text{for each pair of states } y, y'.$$

  - Accept y' as new state, $y_{n+1} = y'$, with probability

$$\min\left(1, \frac{\tilde{\pi}(y')}{\tilde{\pi}(y_n)}\right)$$

  - Otherwise stay at current state, $y_{n+1} = y_n$.
- Demonstration.



Metropolis-Hastings Algorithm

- Hastings generalized Metropolis' algorithm to use asymmetric proposals.
- Suppose the current state is y_n.
  - Propose next state y' from a proposal distribution $q(y'|y_n)$.
  - Accept y' as new state, $y_{n+1} = y'$, with probability

$$\min\left(1, \frac{\tilde{\pi}(y')q(y_n|y')}{\tilde{\pi}(y_n)q(y'|y_n)}\right)$$

  - Otherwise stay at current state, $y_{n+1} = y_n$.

- Practical: Check that detailed balance still holds.

Gibbs Sampling

- State space of Markov chain can often be multi-dimensional, $Y = (Y^{(1)}, Y^{(2)}, \dots, Y^{(m)})$.
- Rather than updating all m variables at once, we can update one variable at a time.
  - Pick a variable j uniformly from $\{1,\dots,m\}$.
  - Compute the conditional distribution $p(Y^{(j)} \mid Y^{(-j)} = y^{(-j)})$.
  - Sample $y^{(j)} \sim p(Y^{(j)} \mid Y^{(-j)} = y^{(-j)})$.
  - Leave the states of all other variables unchanged.
- We can update subsets of variables as well.
- Demonstration.

Practical

Sampling from Exponential with Metropolis

- In this exercise you will implement a Metropolis sampler.
- Use as target an Exponential distribution with λ=1.

$$\pi(y) = \begin{cases} \lambda \exp(-\lambda y) & \text{if } y \geq 0 \\ 0 & \text{if } y < 0 \end{cases}$$

- Use as proposal a Normal distribution centred at current state y, and standard deviation sd

$$q(y'|y) = \frac{1}{\mathrm{sd}\sqrt{2\pi}} \exp\left(-\frac{(y'-y)^2}{2\mathrm{sd}}\right)$$

- Use ex1_mh_exp.m as your MATLAB template, or ex1_mh_exp.R as your R template.

Gibbs Sampling as Metropolis Algorithm

- Gibbs sampling can be understood as a particularly simple case of Metropolis algorithm.
- The proposal distribution is given by:

$$q(y'|y) = \sum_{j=1}^{m} \frac{1}{m} \pi((y')^{(j)} | Y^{(-j)} = y^{(-j)}) \delta((y')^{(-j)} = y^{(-j)})$$

- Suppose y and y′ differ only in dimension j. Then the acceptance probability is:

$$\begin{aligned} & \min\left(1, \frac{\pi(y')q(y|y')}{\pi(y)q(y'|y)}\right) \\ = & \min\left(1, \frac{\pi(y^{(-j)})\pi((y')^{(j)}|y^{(-j)})\frac{1}{m}\pi(y^{(j)}|y^{(-j)})}{\pi(y^{(-j)})\pi(y^{(j)}|y^{(-j)})\frac{1}{m}\pi((y')^{(j)}|y^{(-j)})}\right) = 1 \end{aligned}$$

Practical

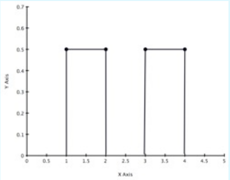
Sampling from Exponential with Metropolis

- Use your function to create three instances of your Markov chain, with 1000 iterations each, start position 1, and standard deviation 1.
- Plot the movement of the three chains, and a histogram of the values they take. Do the results look similar?
- Use your MCMC samples to estimate the mean of the exponential distribution. What is the estimated mean and standard error? Does it agree with the true mean 1? Does it improve if you increase the number of iterations?
- What is the effect of changing the standard deviation?

Practical

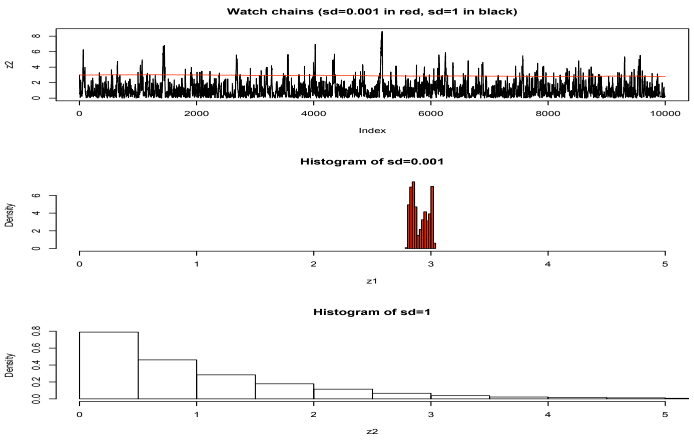
Bimodal Distribution

- Modify your target distribution from the previous practical, from exponential to a bimodal distribution which is 0.5 on [1,2] and on [3,4] but zero elsewhere. The rest of your implementation should work without change.
- Create a Markov chain with 1000 steps, with starting position 3, and standard deviation 1. Plot the chain and a histogram of values taken.
- Do you get the same result if you repeat the procedure?
- What happens if you change the start value to 1?
- What happens if you start at 1, and have sd=0.1?
- What happens if you start at 3 and have sd=0.1?



Watching Your Chain

- Proposal with sd=.001 (red), acceptance rate >99%
- Proposal with sd=1 (black), acceptance rate 52%



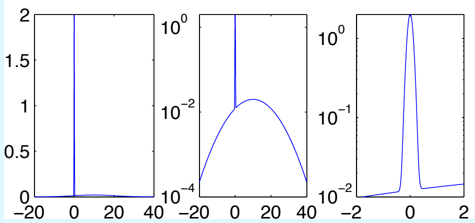
Practical

Mixture of Gaussians

- Using the same code as before, modifying your target distribution to be a mixture of two Gaussians (and maybe the plotting functions):

$$\pi(x) = \frac{.5}{\sqrt{2\pi 100}} e^{-\frac{1}{200}(x-10)^2} + \frac{.5}{\sqrt{2\pi .01}} e^{-\frac{1}{.02}(x)^2}$$

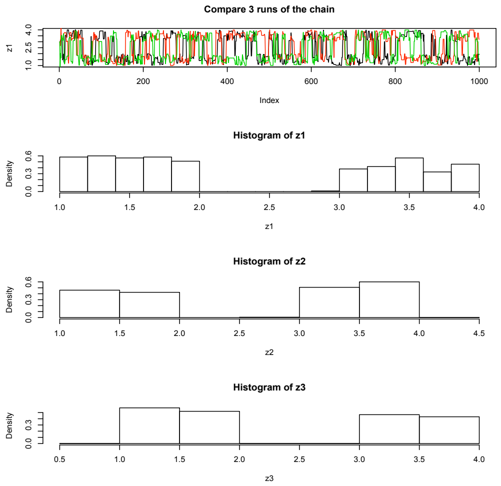
- What proposal sd would you use for good MCMC mixing? Demonstrate using MCMC runs with different sd's, that no single sd gives good mixing.
- Try "fixing" your MCMC run, so that it alternates between two types of updates:
  - ones with sd=10, and
  - ones with sd=.1.



Density Log density Zoomed in

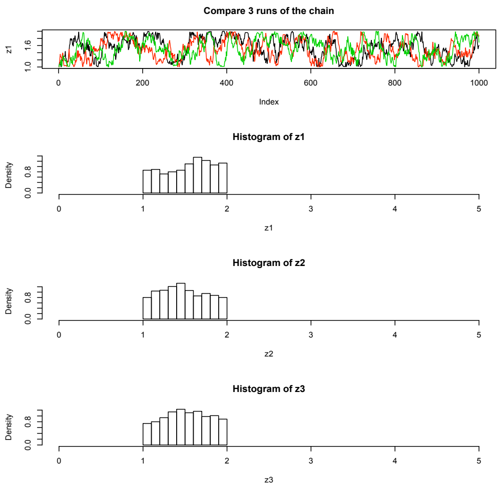
Bimodal Distribution

- Uniform prior on [0,1].
- Proposal distribution is normally distributed around current position, with sd=1.



Bimodal Distribution

- Uniform prior on [0,1].
- Proposal distribution is normally distributed around current position, with sd=.1.



Diagnosing Convergence

- Graphical checks, “eye-balling” the behaviour of the Markov chain.
- Compare estimators obtained from multiple runs from different initial conditions.
- The efficiency of the chain can be measured in terms of the variance of estimates obtained by running the chain for a short time
- There are no guarantees.

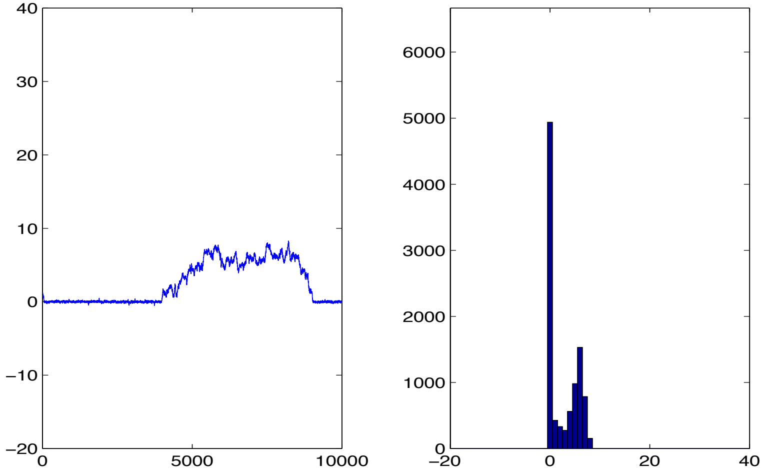
Convergence

- If well constructed, the Markov chain is guaranteed to have the posterior as its stationary distribution.
- But this does not tell you how long you have to run it to convergence.
  - The initial position may have a big influence.
  - The proposal distribution may lead to low acceptance rates.
  - The chain may get caught in a local maximum in the likelihood surface.
- We say the Markov chain mixes well if it can
  - reach the posterior quickly, and
  - moves quickly around the posterior modes.

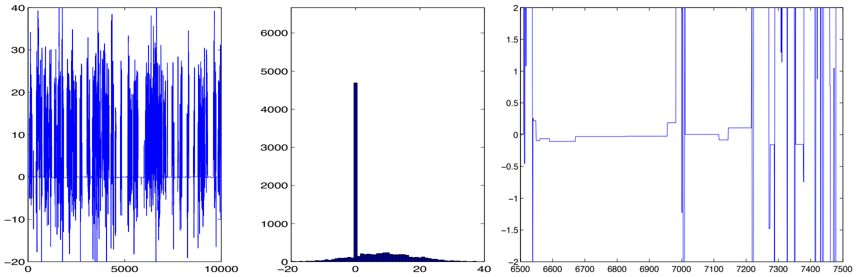
Burn-in

- Often start the chain far away from the target distribution.
  - Target distribution unknown.
  - Visual check for convergence.
- The first “few” samples from the chain are a poor representation of the stationary distribution.
- These are usually thrown away as “burn-in”.
- There is no theory telling you how much to throw away, but better to err on the side of more than less.

Mixture of Gaussians; sd=.1



Mixture of Gaussians; sd=10



Tricks of the Trade

Using Multiple MCMC Updates

- Provided each MCMC update satisfies detailed balance, we can combine different MCMC updates.
- For example we could have one update that proposes huge jumps across our landscape, and another that does detailed local exploration.
- Useful if we expect our landscape to have many narrow maxima.
- Choice of updates can be stochastic, but cannot depend on state of variables that are potentially updated (important!).

Getting it Right

- Geweke (2004) Getting It Right. JASA 99:799-804.
- You have written code that ostensibly gets you samples of posterior distribution $p(y|x)$.
- How do you verify correctness of code?

- Idea: use your code to try generating from prior, and test to see if it does generate from prior.
- Write two additional pieces of easy code:
  1. Sample from prior $p(y)$.
  2. Sample from data $p(x|y)$.

Numerically stable computations

- Almost always better to compute **log probabilities**.
- “log-sum-exp” trick:
  - Have log probabilities $L(i)$=log $p(i)$.
  - Want to compute s = log $\Sigma_i\, p(i)$ = log Σ_i exp($L(i)$).
  - Numerically unstable:
    - s = log(sum(exp(L)))
  - Better:
    - m = max(L)
    - s = m + log(sum(exp(L-m)))

Getting it Right

- Construct a Gibbs sampler for (X,Y) together:
  - $y_{n+1} \sim T(y|y_n,x)$
  - $x_{n+1} \sim p(x|y_{n+1})$
- Compare against:
  - $y_{n+1} \sim p(y)$
  - $x_{n+1} \sim p(x|y_{n+1})$

- You can eyeball histograms, or use Kolmogorov-Smirnov test.
- If estimates obtained from samples differ, something is wrong.

Integrated Autocorrelation Time and Effective Sample Size

- Sequence of states sampled by MCMC is dependent.
- Estimate number of samples before independence.
- Assuming $\mathbb{E}[f(y)] = 0$,

$$\mathbb{V}\left[\frac{1}{N}\sum_{n=1}^{N} f(y_n)\right] = \mathbb{E}\left[\left(\frac{1}{N}\sum_{n=1}^{N} f(y_n)\right)^2\right]$$
$$= \frac{\mathbb{V}[f(y)]}{N}\left(1 + 2\sum_{n=1}^{N-1}\left(1-\frac{n}{N}\right)\frac{C_n}{C_0}\right)$$

- where $C_n = \mathbb{E}[f(y_t)f(y_{t+n})]$ is the lag n autocorrelation.
- As $N \to \infty$, the term in parentheses is:

$$1 + 2\sum_{n=1}^{\infty}\frac{C_n}{C_0}$$

Posterior Predictive Distribution

- Given our posterior distribution on parameters, we can predict the distribution of future data by sampling parameters from the posterior, and simulating data given those parameters.
- We can also verify whether the predictive distribution is consistent with a subset of the data which was held out from inference (a "test" set).
- The Posterior predictive distribution is a useful source of goodness-of-fit testing: if the data we simulate does not look like the data we originally collected, the model is poor.

Practical

Beta-Binomial Model and Allele Frequency (*)

- Use a uniform prior U[0,1] for *r*, and a likelihood which is

$$r^{2n_{AA}}(2r(1-r))^{n_{Aa}}(1-r)^{2n_{aa}}$$

- Write a Metropolis-Hastings MCMC routine to sample from the posterior distribution of *r*.
- Your MCMC function should take a number of iterations, start value and standard deviation as input arguments.
- Try out your new function with n_{AA}=50, n_{Aa}=21, and n_{aa}=29. Use 1000 iterations, a starting value for p of 0.5, and a standard deviation of 0.1.

Practical

Beta-Binomial Model and Allele Frequency (*)

- A standard model used to study the evolution of populations is the Hardy-Wright model, where we assume there is random mating within the population, no selection, and a fixed population size. Each generation, a new set of individuals is born. Suppose we are interested in a gene which has two versions (alleles), A and a where p is the population frequency of allele A . A consequence of this model is that the genotypes AA, Aa and aa will have frequencies r^2, $2r(1 - r)$, and $(1 - r)^2$.
- Suppose we sample n individuals, and find N_{AA} , N_{Aa} and N_{aa} individuals with genotypes AA, Aa and aa respectively.

Practical

Allele Frequency and Inbreeding (*)

- Suppose we have sampled from two populations, one of which is slightly inbred (e.g. due to geographical isolation). This will result in an excess of homozygotes compared with what we expect under Hardy-Weinberg equilibrium.
- One way to capture this is with an inbreeding coefficient f, and assume the genotypes AA, Aa and aa have frequencies $fr+(1-f)r^2$, $(1-f)2r(1-r)$, and $f(1-r)+(1-f)(1-r)^2$ respectively.

Practical

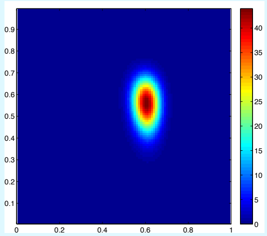
Allele Frequency and Inbreeding (*)

- Modify your likelihood function from previous practical to use the above genotype frequencies.
- Assume independent uniform priors on *f* and *r* on [0,1].
- Write an MCMC routine to sample from the joint distribution of *f* and *r*. Your target is again the product of the prior and likelihood, and your proposal distribution is as before (normal with mean at the current position and standard deviation an input to the routine).
- To avoid numerical problems, modify your likelihood function to be a log likelihood function. Therefore your acceptance condition must change to match this. To be explicit: first check if log(α) > 0. If yes, accept the move. If no, take a U[0,1] sample and if this is less than α then accept, otherwise reject.

Practical

Allele Frequency and Inbreeding (*)

- Now try your function out, for n_{AA}=50, n_{Aa}=21, and n_{aa}=29.
- Use 1000 iterations, standard deviation of 1 (for both f and p), and starting values of f=0.4, and p=0.2. Is the Markov chain mixing well?
- Now drop the standard deviations to 0.1. Is it mixing well now?



Posterior distribution

Practical

Decrypting Messages using MCMC (**)

- You have an English text that has been encrypted by mapping each character to a (usually) different one. For example:

$$a \to s$$
$$b \to !$$
$$c \to \langle\text{space}\rangle$$

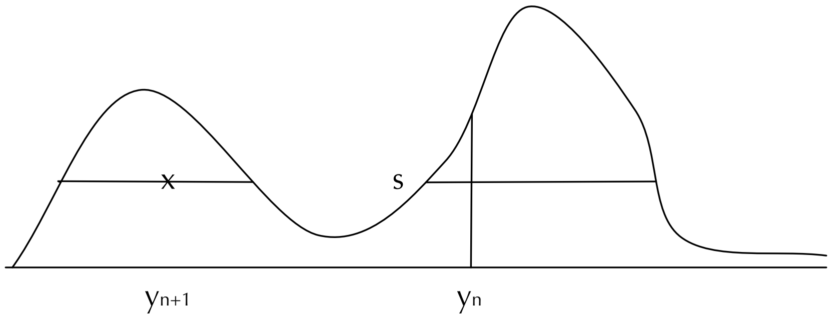
- A text like 'a boy...' might be encrypted as 's3!do...'.
- Assume that each symbol is mapped to a unique symbol. There are 96 symbols given by the text2states and states2text functions.
- Decoding the message by brute force is impossible, so we use MCMC!
- The state of the system consists of a permutation σ of the characters. For example, σ(a) = s, σ(b) = !, σ(c) = <space> etc.

Practical

Decrypting Messages using MCMC (**)

- We model English text using the Markov model from yesterday, where the transition probability matrix has been estimated from another text.
- Derive the likelihood of the encrypted text $e_1e_2...e_m$ given the permutation σ. We use a uniform prior over permutations.
- Derive a Metropolis-Hastings algorithm where a proposal consists of picking two characters and swapping the characters that map to them.
- Implement your algorithm, and run it on the encrypted text in message.txt.
- Report the current decryption of the first 60 symbols every 1000 steps.
- Hint: it helps to pick initial state intelligently and to try multiple times.

Slice Sampling



Hamiltonian Monte Carlo

- Videos.

- We can simulate differential equations by discretizing time.
- This introduces errors, which is corrected by treating the whole procedure as a Metropolis-Hastings proposal, and accepted/rejected.
- Leapfrog discretization:

$$\begin{aligned} v(t+\tfrac{\epsilon}{2}) =& v(t) - \frac{\epsilon}{2}\,\frac{\partial E(y(t))}{\partial y} \\ y(t+\epsilon) =& y(t) + \epsilon v(t+\tfrac{\epsilon}{2}) \\ \hat{v}(t+\epsilon) =& v(t+\tfrac{\epsilon}{2}) - \frac{\epsilon}{2}\,\frac{\partial E(y(t+\epsilon))}{\partial y} \end{aligned}$$

- Volumn preserving, reversible, and almost preserving energy.

Hamiltonian Monte Carlo

- Typical MCMC updates can only make small changes to the state (otherwise most updates will be rejected). -> random walk behaviour.
- Hamiltonian Monte Carlo: use derivative information in log π(y) to avoid random walk and help explore target distribution efficiently.
- Augment state space with an additional momentum variable v:

$$\pi(y,v) \propto \exp(-E(y) - K(v)) \qquad E(y) = -\log\tilde{\pi}(y)$$
$$K(v) = \tfrac{1}{2}\|v\|^2$$

- Hamiltonian dynamics: ball rolling on a frictionless surface.

$$\frac{dy_i}{dt} = \frac{\partial K(v)}{\partial v} = v \qquad \frac{dv}{dt} = -\frac{\partial E(y)}{\partial y}$$

- Total energy is conserved, so

$$\pi(y(t),v(t)) = \pi(y(0),v(0)) \qquad y(0) = y, v(0) = v$$

Other MCMC Ideas

- Data augmentation.
- Rao-Blackwellisation.
- Neal (2003) Slice Sampling. Annals of Statistics 31:705-767.
- Annealing and multicanonical methods (see Iain Murray's PhD thesis).
- Hamiltonian MCMC (see Neal (2010) Handbook of Markov Chain Monte Carlo article).
- Doucet, de Freitas and Gordon (2001) Sequential Monte Carlo in Practice.
- Andrieu, Doucet and Holenstein (2010) Particle Markov Chain Monte Carlo Methods. JRSSB 72:269-342.
- Green (1995) Reversible-jump MCMC. Biometrika 82:711-732.

Further Readings

- Markov Chain Monte Carlo in Practice, 1996, eds Gilks, Richardson, Spiegelhalter.
- Bayesian Data Analysis, 2004. Gelman, Carlin, Stern and Rubin.
- Monte Carlo Strategies in Scientific Computing, 2001, Liu.
- Monte Carlo Statistical Methods, 2004/1999, Robert and Casella.
- Chris Holmes' short course on Bayesian Statistics:
  - http://www.stats.ox.ac.uk/~cholmes/Courses/BDA/bda_mcmc.pdf