Contents

1 Markov chains I

`http://192.168.50.201/statcomp/mcmc/mc1.html`

1.1 Introduction

Consider a game of Ludo. There are a number of counters that move over a board. You think of the counters as variables, their positions being their values, and the board as the set of all possible values. There is a die that determines the moves. Each move causes a counter to change its place based on the roll of the die. Consider only a single counter. Let its position after the i-th move be $X(i)$ (which is a place on the board). $X(i+1)$ is determined from $X(i)$ and a roll of the die. Because of the die the $X(i)$'s are all random. However, the randomness enters in $X(i+1)$ only from two sources: the randomness in $X(i)$and the fresh roll of the die. Thus given $X(i)$ the rv $X(i+1)$ is indep of all the past moves.

Defn

> A *discrete time Markov Chain* means a sequence of random variables $X(0), X(1), ...$ (all defined on some common probability space) such that for any i the rv $X(i+1)$ is indep of $X(0), ..., X(i-1)$ given $X(i)$.

An intuitive way to think about Markov chains is as follows. It is a game with a sequence of random moves, where each of the random moves are independent of others. The current state is the cumulative effect of all the moves so far. In the game of Ludo the random moves are provided by fresh rollings of the die. In general, we call this random move scheme as the *transition kernel*, K. The transition kernel can be thought of as a family of distribution K_x. If $X(i) = x$, then we generate $X(i+1)$ from K_x.

Example: In the Ludo example

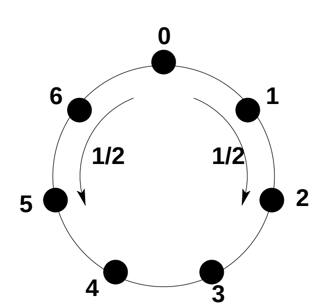
$$K_x(y) = 1/6 \text{ if } y = x+1, x+2, ..., x+6$$

♡

Example: In a simple symmetric random walk over 7 points $\{0, 1, 2, 3, 4, 5, 6\}$ arranged in a circle,

$$K_x(y) = 1/2 if y = x+1 \text{ or } x-1,$$

where the addition and subtraction are modulo 7.



♡

Example: In the last two examples we had discrete kernels. Here is a kernel with density.

$$K_x(y) = N(x, 1)$$

Note that if $X(.)$ is Markov chain with this kernel and $X(0)$ is 0, then $X(n)$ =sum of n iid $N(0,1)$ rvs. ♡

Example: (Iterated Function System or IFS) Computer scientists are fond of IFS's. Here is a simple but interesting example. Our state space is $\mathbf{R}^2$. So each $X(i)$ is a point in the plane. We move from $X(i)$ to $X(i+1)$ as follows.

There are 4 functions f_k from $\mathbf{R}^2$ to $\mathbf{R}^2$ of the form

$$f_k(x) = A_k x + b_k,$$

where A_k's are known 2 by 2 matrices and b_k's are known 2 by 1 vectors. Each f_k has a probabilityp_k attached with it. To move from $X(i)$ to $X(i+1)$ we pick one of the f_k's at random (with probability p_k) and move to $f_k(X(i))$, ie,

$$X(i+1) = f_k(X(i)).$$

These Markov chains are very useful for data compression. To see how, download a little Matlab code from here[1] and run it. All that it does is to run an IFS (the A's and b's are given in the Matlab code) for 10000 steps and plots all the 10000 X's. Is the resulting picture familiar? Here the 4 A's and b's store the picture in a compressed form. ♡

To specify a Markov chain you need to specify three things:

1. The state space, S. This is where all the $X(i)$'s are going to take values. For a ludo it is the board.
2. The transition kernel, K. For a ludo this kernel is determined by the die.
3. The initial distribution, ie the distribution of $X(0)$. This is what sets the ball rolling. In a ludo, this is the distribution degenerate at 0.

1.2 Transition matrix

If S is finite (and so K is discrete) then each K_x is just a probability vector on S. We may stack these on top of one another to get the *transition matrix.* Each of its rows is a probability vector. Such a matrix is called a *stochastic matrix.* The transition matrix is a very useful thing for theoretical purposes. But it takes a lot of place to write/store. So we shall not make much use of the matrix form in this class. Also the matrix form does not exist if S is inifinite. Here is a useful result :

Theorem

If $p(i)$ is a row vector denoting the distribution of $X(i)$ then

$$p(i+1) = p(i)K.$$

1.3 Using the computer

Simulating Markov chain on the computer is quite easy once you know how to make the random moves. Let us write a Matlab code that accepts a general kernel and makes n moves using it.

MATLAB

```
function x = mc(randomMove, start, n)

  x = zeros(1,n);
```

[1]There is a link here in the online notes.

```
x(1) = feval(start);
for i=2:n
  x(i) = feval(randomMove, x(i-1))
end
```

The Matlab function `feval` evalutes a function that is specified by its name. Now let's write a sample kernel.

MATLAB

```
function nextVal = ludoMove(currentVal)
   dieRoll = unidrnd(6);
   nextVal = currentVal + dieRoll;
   if nextVal >= 100
     nextVal = 100;
   end
```

And here is the initial distribution.

MATLAB

```
function startVal = ludoStart
  startVal = 0;
```

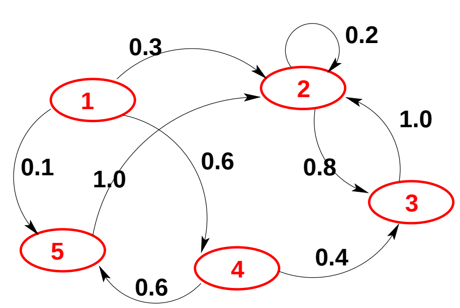
Then it is easy to simulate say 50 steps of this chain as

MATLAB

```
>> x = mc('ludoMove','ludoStart',50);
>> plot(mc('ludoMove','ludoStart',50),'r');
>> plot(mc('ludoMove','ludoStart',50),'g');
>> plot(mc('ludoMove','ludoStart',50),'b');
```

1.4 Transition graph

A transition graph is a valuable tool to understand the structure of a markov chain. It is a directed graph with the states as the nodes. The transition probabilities p_{ij} is shown beside the directed edge from i to j.
Example: Here is a transition graph.



It has 5 nodes, labelled 1 to 5. So the state space is of size 5. The transition matrix, therefure, is a 5 by 5 matrix that we may read off from the graph:

$$\begin{bmatrix} 0.0 & 0.3 & 0.0 & 0.6 & 0.1 \\ 0.0 & 0.2 & 0.8 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.4 & 0.0 & 0.6 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}.$$

Note that the probabilities of all the outgoing arrows from a node add up to 1. The same need not be true for the *incoming* arrows. Also, some node may have arrows pointing to itself, eg, node 2 above. ♡

For small Markov chains transition graphs usually provide more insight than transition matrices. For instance, given the distribution of X_0 you can easily compute the distribution of X_1 using the graph as follows.

Suppose we want to find $P(X_1 = i)$ for any state i. Then consider all the states that lead to i. Let j be a typical such state, and let the edge from j to i have transition probability k_{ji}. Then

$$P(X_1 = i) = \sum_{j \sim i} (k_{ji} * P(X_0 = j),$$

where the sum is over all j's adjacent to i. This formula holds because, in order for X_1 to be at state i, we need X_0 to be at some state j leading to i. The chance of moving from j to i is k_{ji}.

Exercise: Consider the transition graph given above. Suppose X_0 is distributed uniformly over the state space $1, 2, 3, 4, 5$. Compute the distribution of X_1. ♡

1.5 Markov chain distributions

There are three important types of distributions connected with a Markov chain:

- finite stage distributions
- stationary distribution
- limiting distribution

1.6 Finite stage distribution

After k steps the chain is at $X(k)$, which is a random variable. Let $p(k)$be the distribution of $X(k)$. These are the finite stage distributions. $p(0)$ is the initial distribution, and hence is part of the specification of the Markov chain.

Example: Consider the simple symmetric random walk example from above. Let $p(k,i)$ denote the i-th entry of the probability vector $p(k)$. Then

$$p(k+1,i) = 0.5 * (p(k,i-1) + p(k,i+1)).$$

♡

1.7 Stationary distribution

Let us continue with the last example.

Suppose that $p(0)$ is unif, ie, $p(0,i) = 1/7$ for all i. Then it is easy to see that $p(k)$ is unif for any k. Thus, if the Markov chain starts with unif distribution it remains "stationary" at that distribution. *This does not mean that $X(k)$'s do not change. It means that $p(k)$'s do not change.*

Defn

Suppose that L is a distribution such that if $p(0) = L$, then $p(1) = L$. Then we call L a *stationary distribution* for the Markov chain.

Note:

- If $p(0) = L$ and L is stationary, then $p(k) = L$ for all k
- If $p(k) = L$ then $p(k+1) = L$
- If we have a finite state space Markov chain then $LK = L$, ie, L is a (left) eigenvector of K corresponding to eigenvalue 1.

Exercise: Is discrete uniform the only stationary distribution for the above Markov chain? ♡

1.8 Limiting distribution

As k goes to infinity (ie, as we run the Markov chain for a long time) p(k) may or may not converge. To talk about convergence we need a metric to measure the distance between probs on S. If S is finite, then a probability is just a finite-dimensional vector, and we may use Euclidean metric. In general, we use the following metric.

Defn

Let p and q be probabilities on the same space. Then the *total variation distance* between them is

$$d(p,q) = sup\{|p(A) - q(A)| : A \text{ measurable}\}$$

Check that it is indeed a metric, ie,

1. $d(p,q) \geq 0$for all p,q,and $d(p,q) = 0$ iff $p = q$
2. $d(p,q) = d(q,p)$
3. $d(p,q) \leq d(p,r) + d(r,q)$ for any three probabilities p,q,r

Exercise: Check that it is *equivalent* to Euclidean metric for finite S. ♡

Defn

If there is a distribution q on the state space such that $d(p(k),q)$ goes to 0 as k tends to infinity, we call q a *limiting distribution.*

Note that limiting distribution need not be unique. It may very well depend of the initial distribution.

Exercise: Can you construct such an example? [Hint: It's very easy ;-)] ♡

1.9 Using the computer

Getting an idea about $p(k)$ will be important for our purpose later. It can be obtained as follows. Run the Markov chain for k steps to get one realisation of $X(k)$. Next run the chain again *independently from the very beginning* for k steps to get another realisation of $X(k)$. Go on like this N times. Plot a histogram of these Nrealisations. Here is a Matlab code to to this:

MATLAB

```
data = zeros(100,10);
for i = 1:100
  x(i) = mc('kern','start',50);
  data(i,:) = x(1:5:50);
end

for i=1:10
  subplot(5,2,i);
  frq = histc(data(:,i),1:100);
  bar(frq./100); axis([0 101 0 1]);
end
```

Matlab has a function called `hist` to draw histograms. But it labels the axes differently for different histograms making comparison difficult. It is possible to mend that by putting the labels manually, but that's cumbersome. So we have used `histc` to find the frequencies and `bar` to plot.

Computing the total variation distance is actually very easy in Matlab. You can compute it in just one line. Here's a hint: If `x,y,z` are vectors of the same length in Matlab then `x(y>z)` denotes the subvector of `x` consisting of all those `x(i)`'s for which `y(i)>z(i)`.

2 Markov chains II

`http://192.168.50.201/statcomp/resample/mc2.html`

2.1 Relation between the distributions

The following result is intuitive, but it is less obvious how to *prove* it. Give it a try!

Theorem

> A limiting distribution must be a stationary distribution. The converse is not always true, even if the Markov chain has unique stationary distribution.

Exercise: Get a counter example for the converse. $\heartsuit$

2.2 Using the computer

Let us consider the circle example from the last page. Suppose `S` is of size 5. Start from 0. Plot `d(p(k),u)` against `k`, where `u` is the uniform distribution. Repeat this for `|S| = 6`. Can you explain the difference?

A transition graph is a valuable tool to understand the structure of a markov chain. It is a directed graph with the states as the nodes. The transition probabilities p_{ij} is shown beside the directed edge from i to j.
Example: Here is a transition graph.

It has 5 nodes, labelled 1 to 5. So the state space is of size 5. The transition matrix, therefure, is a 5 by 5 matrix that we may read off from the graph:

$$(0.0 \quad 0.3 \quad 0.0 \quad 0.6 \quad 0.10.0 \quad 0.2 \quad 0.8 \quad 0.0 \quad 0.00.0 \quad 1.0 \quad 0.0 \quad 0.0 \quad 0.00.0 \quad 0.0 \quad 0.4 \quad 0.0 \quad 0.60.0 \quad 1.0 \quad 0.0 \quad 0.0 \quad 0.0).$$

Note that the probabilities of all the outgoing arrows from a node add up to 1. The same need not be true for the *incoming* arrows. Also, some node may have arrows pointing to itself, eg, node 2 above. $\heartsuit$

For small Markov chains transition graphs usually provide more insight than transition matrices. For instance, given the distribution of X_0 you can easily compute the distribution of X$_1 using the graph as follows$.

Suppose we want to find $P(X_1 = i)$ for any state i. Then consider all the states that lead to i. Let j be a typical such state, and let the edge from j to i have transition probability k_{ji}. Then

$$P(X_1 = i) = \sum_{j \sim i} (k_{ji} * P(X_0 = j)),$$

where the sum is over all j's adjacent to i. This formula holds because, in order for X_1 to be at state i, we need X_0 to be at some state j leading to i. The chance of moving from j to i is k_{ji}.
Exercise: Consider the transition graph given above. Suppose X_0 is distributed uniformly over the state space $1, 2, 3, 4, 5$. Compute the distribution of X_1. $\heartsuit$

2.4 Some theory

When designing a MCMC algorithm to simulate (approximately) from a distribution p with support S, we try to get hold of a Markov chain with state space S, such that for every the initial distribution, the Markov chain will converge to p. How do we know if a Markov chain has this property or not? We present below some relevant results from classical Markov chain theory.

There is a rich theory of Markov chain, even when S is a finite set. While most the course will involve Markov chains with state space $\mathbf{R}^n, weshallpresentthetheoryonlyforfinitestatespaceMarkovchains.Thereasonisthatboththetheoriesarethesameinspirit, thoughthe\mathbf{R}^n$ case has lots of extra subtleties to make things rigorous. In any case, the theory merely gives us a set of necessary and sufficient conditions of when a Markov chain will converge. Some of these are difficult to check in real problems.

2.4.1 Irreducibility

Example: Consider a Markov chain whose transition graph has two disjoint components. Do you think that it can have a unique limiting distribution irrespective of initial distribution?

No, because you can never leave the component you started in. $\heartsuit$

The same problem may sometimes occur in more subtle forms even when the transition graph is in one piece.

Example: The transition graph shown earlier is in one piece. But notice that you cannot move from 4 to 1. So if we start at 4, then the limiting distribution, even if one exists, cannot include 4 in its support. $\heartsuit$

To preclude such situations we need the following definition.

Defn

A Markov chain is called *irreducible* if you can arrive from any state to any state (you may need more than one step to do so.)

The examples above were not irreducible. Such Markov chains are not good candidates for MCMC. This is an important condition in MCMC, because our aim is to start from one state and then keep on running the Markov chain so that we visit all the states with appropriate probailities. We do not want to get "stuck" in any subset of the state space.

2.4.2 Aperiodicity

Example: Consider a Markov chain that is just a 3-cycle.

$$1 \to 2 \to 3 \to 1,$$

each transition happening with probability 1. Do you think it is a good candidate for MCMC?

No. While it does have unique stationary distribution (find it!), if you start from 1, say, you'll move in a periodic way, never converging anywhere. Here is a more formal way of seeing this.

Let, if possible, the Markov chain have a unique limiting distribution, p, irrespecitve of the intial distribution. Pick any x such that $p(x) > 0$. Start the chain at x, ie, $P(X_0 = x) = 1$. What is the chance that $X_n = x$ if n is not divisible by 3? It is zero, because of the 3-cycle nature of the Markov chain. So clearly it cannot converge to p, since $p(x) > 0$. $\heartsuit$ The above argument tells us that if we have a Markov chain where some state can be returned to only in multiples of some $h > 1$, then it is not a good candidate for MCMC. Such Markov chains need not always look like a single cycle.

Example: Suppose that we have two cycles connected at a single vertex. One cycle is of size 4, the other of size 6. Do we have the same problem here?

Yes, since we can return to the common vertex only at multiples of 2. Actually, we should better say that we *cannot* return to 1 at *non-multiples* of 2. $\heartsuit$

Such states in a Markov chain are called periodic. Here is the definition.

Defn

A state, x, is called *periodic* of period h if you can return to it only in multiples of h. (h is the smallest such number). More formally, let B be the set of all integers such that you can return to x at these steps. Then the period of x is $$h = gcd(B).$$

Example: (contd) In the last example find period of 1. We can come back to 1 only in multiples of 4 or 6. So B consists of multiples of 4 and 6. Hence $gcd(B) = gcd(4, 6) = 2$. ♡

A Markov chain with at least one state having period more than 1, is not a good candidate for MCMC.

Example: Consider the following transition graph. Let the periods of i and j be d_i and d_j. What can you say about the relation between them?

Note that we can return to i in 8 steps (5 dashed steps followed by 3 solid steps.) So d_i must divide 8. Also, the same argument says that $d_j$divides 8, as well. Let B_i be the set of all numbers d such that you can go from i to i in exactly d steps. Now pick any number t in B$_i.Sothereissomeloopfromitoioflengtht.Clearly,wenowhavereturnpathtojoflength$8+t$(solid,dottedthendashed).Sod_j$ divides $8+t$, ie, d_j divides t. But t in B_i was arbitrary, so d_j divides every member of B_i. Hence, d_j divides d_i. By symmetry, d_i divides d_j, as well.

So $d_i = d_j$. ♡ Similar argument proves the following theorem.

Theorem

For an irreducible chain, each state has the same period.

Defn

If the common period of an irreducible chain is 1, then we say that the chain is *aperiodic*.

2.4.3 Some theorems

So far we have some conditions necessary for a Markov chain to be good candidate. Here is a sufficent condition.

Theorem

An irreducible, aperiodic Markov chain with finite state space has only one stationary distribution, which is also the unique limiting distribution.

Analogous results hold for Markov chains with (uncountably) infinite state spaces also, if they satisfy a condition called *(Harris) recurrence*. There is a good amount of literature defining and characterising recurrence for general Markov chains. However, those results are rarely useful for MCMC, since the conditions of those theorems are usually too difficult to be checked in practice. So we shall not go into them in this course.

3 Markov Chain Monte Carlo (MCMC)

`http://192.168.50.201/statcomp/mcmc/intro.html`

3.1 Introduction

Markov Chain Monte Carlo is a modern computational tool for statisticians. It is a technique to simulate from very complicated distributions. We shall introduce the concept with a simple example.

Example: Suppose that we want to simulate from the probability vector $(3/7, 4/7)$. We shall call it our target distribution. It is like tossing a coin with probability of head 3/7. We already know how to do this simulation easily. However, we shall apply MCMC method to do the simulation.

An MCMC method starts by obtaining a Markov chain whose stationary distribution is the target distribution. Here is the transition matrix for one such Markov chain:

$$P = \begin{bmatrix} 1/3 & 2/3 \\ 1/2 & 1/2 \end{bmatrix}.$$

We shall discuss later how this transition matrix is obtained. Observe that the target distribution is its unique stationary ditribution, i.e., the only solution for

$$\pi P = \pi$$

is $\pi = (3/7, 4/7)$. Also it is the unique limiting distribution of the Markov chain. The state space of the chain is $\{1, 2\}$. Let us run the Marko chain starting from $X(0)$ which is any random variable taking values in this set. Let $X(n)$ be the random variable denoting the n-th step of the Markov chain. Then the distribution of $X(n)$ tends to π as n goes to infinity.

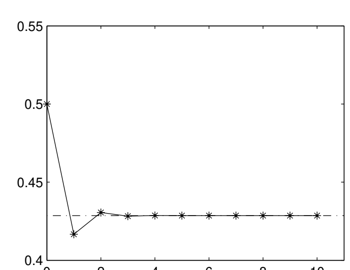
For instance, suppose that the distribution of $X(0)$ is $(1/21/2)$. Then the distribution of $X(1)$ is

$$(1/21/2) \begin{bmatrix} 1/3 & 2/3 \\ 1/2 & 1/2 \end{bmatrix} = (5/12, 7/12).$$

If $p(n)$ denotes $P(X(n) = 1)$, then it is easy to see that

$$p(n+1) = 0.5 - p(n)/6.$$

Clearly, $p(n)$ converges to $3/7$ at an exponental rate. The following plot shows $p(n)$ against n.



Convergence of Markov chain distribution

This shows that the distribution of $X(6)$ is very close to π. So we can run the chain for 6 steps and declare $X(6)$ as our output. Here we throw away $X(0), X(1), ..., X(5)$. This 5 is called the **burn-in period**. ♡

3.2 A homely example

Running a MCMC all the way in order to get just a single realisation may look sheer madness at first. But there is at least one example where we all have been using a MCMC scheme from our childhood!

Consider shuffling a deck of cards. The aim is to generate a uniform random permutation of n=52 cards. Do you ever do that by drawing a SRSWOR? No! Rather, you just shuffle the deck a number of times. This is the same as running a Markov chain. If done properly it takes about 8 shuffles to get a deck random. There is some fascinating math regarding card shuffling. [Details...][2] .

[2]There is a link here in the online notes.

3.3 Points to ponder

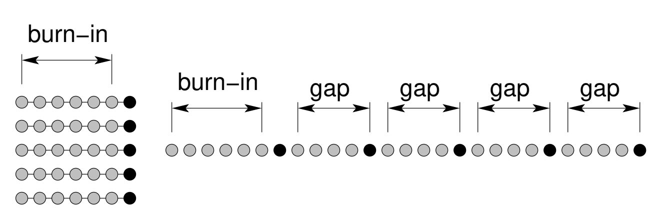
The above method gives rise to the following points:

- **Why take all this trouble?** The reason is there are some complicated target distributions which are difficult to be simulated directly, but for which it is possible to obtain a Markov chain that has the required target distribution as the unique limiting distribution.

- **How to get a Markov chain** that has the required distribution as the unique limiting distribution? There are some standard methods that we are going to learn in this course.

- **How to assess convergence?** This is a tricky question which is still a reasearch topic. There are a number of approaches: explicitly computing a lower bound for a large enough n, using heuristic tests to check if a given value of n is large enough, replacing n by a stopping time. We shall take a look at some of these later.

- **How to generate iid samples** from the target distribution? The method we have outlined above needs to run a Markov chain for 6 steps to get a single sample from the target distribution. To generate 100 iid samples we may repeat the entire process 100 times, each time starting with a fresh simulation of $X(0)$. This is called the **many short chains approach**. Some statisticians prefer the **one long chain approach.** In this approach you run only one Markov chain. After the burn-in period is over, decide upon a gap length k and pick every k-th random variable after that. Thus, if b is the burn-in period then the n-th sample point is $X(b+nk)$.



Many-short-chains versus one-long-chain

No clear conclusion exists as to which approach is better.

3.4 Relation between Markov chain and MCMC

A MCMC algorithm is used to simulate from some given "tough" probability distribution , with density f, say. Designing a MCMC algorithm simply amounts to designing a Markov chain that converges to f *irrspective of the initial distribution.* This last condition frees us from any worry about how to start the Markov chain. Note that this f is then the *unique* limiting distribution to which we *must* converge. This motivates the following definition.

Defn

Call a Markov chain a **good candidate for MCMC** if it has unique limiting distribution irrespective of initial distribution. This terminology, by the way, is not standard.

The following facts will help to explain the point.

- A Markov chain can have unique stationary distribution, but still fail to be a good candidate.

- A Markov chain can have unique limiting distribution, but still fail to be a good candidate. [In this case, we may not converge to anything if we start from certain initial distributions.]

Exercise: Think of examples illustrasting each of these. [Hint: there are simple examples for both.] ♡

3.5 Hunting for good candidates

As we have mentioned earlier, designing a MCMC algorithm is basically getting hold of a suitable Markov chain. Now, obviously we want the Markov chain to be good candidate. So we should better have a large class of good candidates ready at hand to choose from. One such class is the set of all Markov chains that are irreducible, aperiodic and recurrent. All known MCMC algorithms work with only Markov chains from this class of good candidates.

Exercise: Consider a finite connected graph with vertex set V. At least one vertex has a self-loop. We shall run a Markov chain with V as its state space. Let $X(0)$ be a vertex chosen at random (each vertex equally likely.) At step n we are at vertex $X(n)$. Pick one of the edges at random (each edge equally likely) and follow that edge. Show that this Markov chain is irreducible and aperiodic. What is its unique limiting distribution? ♡

4 Gibbs sampler

`http://192.168.50.201/statcomp/mcmc/gibbs1.html`

4.1 Introduction

Gibbs sampling is the most popular MCMC scheme to date. It was invented by statistical physicists, and was called Heat bath algorithm by them. The Geman brothers renamed it to Gibbs sampler.

4.2 Set up

We have a random vector $(X_1, ..., X_k)$ with joint density $f(x_1, ..., x_k)$, that we want to simulate from. Unfortunately, we cannot do so directly due to some reason or other (eg, f is not available in closed form). However, we can simulate from *all* the following conditional distributions:

$$X_i | X_1, ..., X_{i-1}, X_{i+1}, ..., X_k \text{ for } i = 1, ..., k.$$

These are called the **full conditional** distributions. We shall denote the random vector $(X_1, ..., X_{i-1}, X_{i+1})$ as $\mathbf{X}_{-i}$.

Then the Gibbs sampler is the following MCMC scheme:

Gibbs sampler

Start with some initial values $X_1(0), ..., X_k(0)$
For $n = 1, ..., N$
    For $i = 1, ..., k$
        Generate $X_i(n) \sim LAW(X_i | \mathbf{X}_{-i} = \mathbf{X}_{-i}(n-1))$.
    End for
End for
Output $X_1(N), ..., X_k(N)$

Example: (Beta-binomial) Consider two random variables X, Y (X discrete, Y continuous) with joint density[3] given by

$$f(x,y) = ^n C_x y^{x+a-1}(1-y)^{n-x+b-1} \text{ for } x = 0,1,...,n \text{ and } y \text{ in } [0,1].$$

It is easy to see that given $Y = y$, the conditional distribution of X is $Binom(y)$, and given $X = x$, that of Y is $Beta(x+a, n-x+b)$. We want to sample from this beta-binomial distribution. The Gibbs sampler proceeds as follows.

1. Start from some value of (X_0, Y_0).
2. Generate X_i from $Binom(Y_{i-1})$
3. Generate Y_i from $Beta(X_i + a, n - X_i + b)$

♡

Exercise: It is also possible to simulate exactly from the joint distribution. Figure out how. ♡

Exercise: (Autoexponential) Here we have the density

$$f(x,y,z) \propto exp(-x - y - z - axy - byz - czx) \text{ when } x,y,z > 0.$$

Show that $Z|(X,Y)$ is Exponential. However, the marginals and other conditionals are complicated. Set up a Gibbs sampler to simulate from this distribution. ♡

4.3 Stationary distribution

A Gibbs sampler is a Markov chain that has the correct stationary distribution. To show this we shall need the following theorem.

Theorem

> Let X, Y, Z be any three random variables defined on the same probability space. If the conditional distribution of $X|Y$ is same as that of $Z|Y$ then the joint distribution of (X,Y) is same as that of (Z,Y).

Proof:

For notational simplicity we shall assume that X, Y, Z all have joint densities: $f(x)$ for X, $f(y)$ for Y, $f(x|y)$ for $X|Y$, and so on. Then the joint density of (X,Y) is

$$f(x,y) = f(y)f(x|y),$$

and, similarly, the joint density of (Z,Y) is

$$f(z,y) = f(y)f(z|y).$$

But these are the same, since we are given that $f(x|y) = f(z|y)$. ♡

Now we shall show that Gibbs sampler has the correct stationary distribution. Suppose we start with $(X_1(0), ..., X_k(0))$, which already has joint distribution f. If we update $X_1(0)$ to $X_1(1)$, where the latter is drawn from the conditional distribution

$$X_1 | \mathbf{X}_{-1} = \mathbf{X}_{-1}(0),$$

then the theorem guarantees that the two random vectors

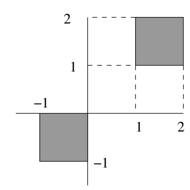
$$(X_1(0), X_2(0)..., X_k(0)) and (X_1(1), X_2(0)..., X_k(0))$$

have the same joint distribution, namely, f. Similarly, for updating the other X_i's. Thus, if we start from f, then after one Gibbs sampler step we shall remain at f.

[3] What does "joint density" mean when X is discrete and Y is continuous?

4.4 Irreducibility

Gibbs sampler need not be irreducible in general.
Example: (disconnected support) Consider generating from a distribution with support which is union of the squares $[-1,0] \times [-1,0]$ and $[1,2] \times [1,2]$. Clearly we can never jump from one to the other using just horizontal or vertical steps.



Gibbs sampler cannot move from one box to the other

♡

Exercise: Is it true that the Gibbs sampler is *always* reducible if the support of the target distribution has more than one components? ♡

Here is a sufficient condition for irreducibility.

Theorem

If positivity of all the marginal densities imply positivity of the joint density, then

- the Gibbs sampler is irreducible,
- for any L_1 function h, we have
$$\sum h(\mathbf{X}_n)/N \rightarrow E(h(\mathbf{X}))a.e.[f].$$
- if the chain is also aperiodic, then it converges to f in total variation norm.

Theorem

(A sufficient condition for irreducibility and aperiodicity of a Gibbs sampler) Suppose that we have the following two conditions:

1. There is $\delta > 0$ such that for
$$\mathbf{x} = (x_1, ..., x_k)$$
$$\mathbf{y} = (y_1, ..., y_k)$$
in the support of f with $|\mathbf{x} - \mathbf{y}| < \delta$, we have
$$f(y_i|y_1, ..., y_{i-1}, x_{i+1}, ..., x_p) > 0 \text{ for all } i.$$
2. There is $\delta' < \delta$ such that any two points $\mathbf{x}, \mathbf{y}$ in the support of f can be connected via a chain of overlapping disks of radius δ'.

Then the Gibbs sampler is irreducible and aperiodic.

4.5 Completion Gibbs sampler

Sometimes we need to simulate from a univariate density f, which is a marginal of some higher dimensional density g. Then we can simulate from g by Gibbs sampler, and keep only the f-part of it.
Example: Suppose that $X \sim N(\theta, 1)$ and $\theta \sim Cauchy(2, 1)$. Then the posterior is proportional to

$$\frac{e^{-(\theta-x)^2/2}}{1+(\theta-2)^2}$$

This may be written as the marginal for $g(\theta, \lambda)$ which is proportional to

$$exp(-(\theta - X)^2/2)exp(-(1 + (\theta - 2)^2)\lambda).$$

Now notice that $\theta|\lambda$ is normal and $\lambda|\theta$ is Exponential.

Note that λ has no physical interpretation at all, it merely helps to set up the Gibbs sampler. ♡

5 A Bayesian excursion

`http://192.168.50.201/statcomp/mcmc/gibbs2.html`

5.1 Introduction

MCMC is popular among Bayesians in the following set up. We have data, model and prior. The Bayesian approach is to compute various quanities based on the posterior. Often the poserior is formidable and is given as a miltidimensional integral upto an unknown normalizing constant. If we cannot tackle it analytically, we need to simulate from the posterior. MCMC is one popular way to do this. Though this is not a course on Bayesian statistics, we provide here a brief introduction to the subject. The ideas presented here will be used at various places in the course.

5.2 Bayes theorem

Bayesian statistics is based on Bayes theorem:

Theorem

If A, B are two events in the same probability space, then

$$P(A|B) = P(A)P(B|A)/P(B)$$

A Bayesian would think of this theorem as follows. A and B are two events that happen *one after another*, first A and then B, eg,

$A = \{$a burglar has entered my house$\}$
$B = \{$my dog barks$\}$

$P(A)$ reflects our belief about how probable an event A is. In this case, P(burglar has entered) is quite low (hopefully!), say it is 0.1. If I have a vigilant dog it is almost sure to bark if a burglar has entered the house. So $P(B|A)$ is high, say 0.9. Of course, there are some dogs who bark all the time for no apparent reason whatsoever! But my dog is not one of them, so $P(B)$ is low, say 0.2.

Now if I hear my dog bark I shall reason as follows:

> Well, my dog does not usually bark, but now he *is* barking. He is almost sure to bark if there's a burglar in the house. So possibly there *is* a burglar in the house right now!

Bayes theorem is the mathematical version of this:

$$P(A|B) = 0.1 \times 0.9/0.2 = 0.45$$

Bayes theorem may also be written in the following form.

Theorem

Let X, Y be two discrete random variables defined on the same probability space. Let their probability mass functions (pmf) be $p(x)$ and $q(y)$, respectively. If $p(x|y)$ and $q(y|x)$ denote the conditional pmf's of X given Y, and Y given X, respectively, then

$$\begin{aligned} p(x|y) &= p(x)q(y|x)/q(y) \\ &= p(x)q(y|x)/\sum_x (p(x)q(y|x)) \end{aligned}$$

Note that the denominator in the last expression is just the sum of the numerator.

Proof:

Take $A = X = x$ and $B = Y = y$. Then $p(x|y)$ is just $P(A|B)$ and $q(y|x) = P(B|A)$. Now apply the last theorem. ♡

Not surprisingly, there is a version of Bayes theorem for random variables with densities. We shall sweep some technicalities under the rug to state the following theorem.

Theorem

Let X, Y be real-valued random variables defined on the same probability space. Let them have densities $f(x)$ and $g(x)$. The conditional densities will be denoted by $f(x|y)$ and $g(y|x)$. Then

$$\begin{aligned} f(x|y) &= f(x)g(y|x)/g(y) \\ &= f(x)g(y|x)/\int f(x)g(y|x)dx \end{aligned}$$

5.3 Bayesian inference

In statistics we are almost always trying to infer about unobservable quantities based on what we *can* observe. It is customary to call the unobservable quantities as **parameters**, and the observable ones as **data**. This is much like the burglar-dog example above: entry of the burglar is unobservable (assuming a clever burglar!) but the barking of the dog is surely observable (or audible, rather!), and we want to infer about the former based on the latter. For this we apply Bayes theorem, which connects three pieces of informations:

1. **Prior:** What we already know about the parameter (eg, burglary is rare, $p(x)$ or $f(x)$)

2. **Model:** A connection between the parameter and the data (eg, the dog is almost sure to bark if there is a burglar, $q(y|x), g(y|x)$)

3. **Posterior:** Updated knowledge about the parameter after taking the data into account (eg, if the dog has barked, then there is possibly a burglar around, $p(x|y), f(x|y)$).

Out of these three, the first two are inputs and the last one is the output from Bayes theorem.

In classical statistics (also called *frequentist* statistics to distinguish it from *Bayesian* statistics) we consider parameters as unknown constants that we want to find out. A Bayesian, on the other hand, considers both the data and the parameters as random variables. His current knowldge about the parameters is summed up in a prior distribution. After taking the data into account he updates this knowledge. The updated knowldege takes the form of another distribution: the posterior distribution. Note that

1. prior is a probability distribution defined on the *parameter space*
2. model is a probability distribution defined on the *sample space*
3. posterior is a probability distribution defined on the *parameter space*

Computing posterior distribution based on prior and model is the main thing in Bayesian statistics.

5.4 Computing posteriors by inspection

One may of course apply Bayes theorem verbatim to compute the posterior. But usually the sum or the integral in the denominator proves too difficult to compute. However, note that the denominator is just the sum or integral of the numerator, and hence is the normalizing constant to make the probability add/integrate up to 1. If the posterior is some standard distribution then this observation allows us to compute the posterior merely by inspection without any explicit summation/integration. For this one needs to remember a list of facts about standard distributions.

Example: Consider the $N(\mu, \sigma^2)$ density:

$$\frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{\mu^2}{2\sigma^2}\right] \times \exp\left[-\frac{1}{2\sigma^2}x^2 + \frac{\mu}{\sigma^2}x\right],$$

which we have split up into two factors: one free of x and the other involving x. Now all that we shall remember about the second factor is its *form* as a function of x. So the form of the density is

$$(\text{stuff free of } x) \times exp(\text{some quadratic in } x).$$

This form *along with the fact that the support of the density is the entire real line* is enough to show that the density is Gaussian. Also, if the quadratic in x is

$$Ax^2 + Bx + C,$$

then it is trivial to back-calculate μ and σ^2 as

$$\sigma^2 = -1/(2A), \mu = -B/(2A).$$

Note that we do not even need to consider C. ♡

I personally like to think of the factor involving x as the *soft* part of the density. In order to compute standard posteriors by inspection you need to remeber the forms of the soft parts and supports for standard distributions.

1.
- **Distribution:** $N(\mu, \sigma^2)$
- **Soft part:**$exp(Ax^2 + Bx)$, where $A < 0$
- **Support:**$(-\infty, \infty)$
- **Parameters:**

2.
- **Distribution:** $Beta(a, b)$
- **Soft part:** $x^p(1-x)^q$, where $p, q > -1$.
- **Support:** (0,1)
- **Parameters:** $a = p + 1$ and $b = q + 1$.

3.
- **Distribution:** Dirichlet$(\alpha_1, ..., \alpha_k)$
- **Soft part:**

$$\prod_{i=1}^{k} x_i^{\alpha_i - 1}$$

- **Support:** $\{(x_1, ..., x_k) : x_i > 0, \sum x_i = 1.\}$

4.
- **Distribution:** Gamma(a, p) where $a, p > 0$
- **Soft part:** $x^{-p}exp(-ax)$
- **Support:** $[0, \infty)$

5.
- **Distribution:** Inverse gamma(a, p) where $a, p > 0$
- **Soft part:** $x^{1-p}exp(-a/x)$
- **Support:** $[0, \infty)$

6.
- **Distribution:** Wishart(Σ, k) where Σ is a m by m positive definite matrix and $k \geq m$ is an integer.
- **Soft part:** $|x|^{(k-m-1)/2}exp(-tr(\Sigma^{-1}x/2))$,
- **Support:** Positive definite matrices

7.
- **Distribution:** Inverse Wishart(Σ, k) where Σ is a m by m positive definite matrix and $k \geq m$ is an integer.
- **Soft part:** $|x|^{-k/2}exp(-tr(\Sigma^{-1}x^{-1}/2))$,
- **Support:** Positive definite matrices

You may like to recall the definitions of the following distributions:

- **Inverse gamma:** We say that a random variable X has inverse gamma(p, λ) distribution if $1/X$ has gamma(p, λ) distribution.
- **Wishart distribution:** Wishart is the analogue of χ^2 for matrix-valued random variables. If $X_1, ..., X_k$ are iid $N_d(0, I)$ then the distribution of $\sum(X_i X_i')$ is called Wishart with k degrees of freedom.
- **Dirichlet:** It is the multivariate analogue of Beta. Let $\alpha_1, ..., \alpha_k$ be any k positive numbers. Then a random vector X is said to have Dirichlet$(\alpha_1, ..., \alpha_k)$ distribution, if it has joint density proportional to

$$\prod_{i=1}^{k} x_i^{\alpha_i - 1}.$$

We shall use the above table to compute standard posteriors by inspection. The support of the posterior is same as that of the prior.

Example: Suppose that we have data

$$X_1, ..., X_n \text{ iid } N(\mu, 4).$$

Our prior for μ is $N(3,1)$. What is the posterior of μ?

Here μ is the object of interest, so anything involving μ is soft, everything else being hard. Collecting soft parts of the model and the prior and multiplying we get the soft part of the posterior as

$$exp(-1/8\sum_i (X_i-\mu)^2) exp(-1/2(\mu-3)^2),$$

which is obviously of the form exp(quaratic in μ). So the posterior is again a normal distribution. Now you'll need to explicitly compute ony the coefficient of μ^2 and μ in order to get the mean and variance of the distribution. Observe that we needed no integration at all. ♡

5.5 Conjugate priors

Even for simple prior and model the posterior may very well be formidable. To avoid this one often chooses the prior and the model from special families of distributions that are guaranteed to produce a posterior in the same family as the prior.

Defn

Let D_1 be a family of distributions on the parameter space, and D_2 be a family of distributions on the sample space. We call D_1 a *conjugate prior family* for D_2 if for any prior in D_1 and any model in D_2, the posterior is again in D_1.

You should check for yourself the following examples of conjugate prior families. For each case work out the posterior.

D_1	D_2
$p \sim$ Beta	Binomial(n,p)
$p \sim$ Dirichlet	Multinomial(n,p)
$\mu \sim$ normal	normal(μ, known var)
$\sigma^2 \sim$ inverse gamma	normal(known mean,σ^2)
$\Sigma \sim$ inverse Wishart	multinormal(known mean,Σ)
$\lambda \sim$ Gamma	Gamma(known α,λ)
$\lambda \sim$ Gamma	Poisson(λ)

5.6 Improper priors

In fact, priors need not be valid probability distributions. All that we need is that the posterior is a valid probability distribution. If the prior is a prob distribution, then the posterior is guaranteed to be one, as well. But the posterior may be a prob distribution even if the prior is not. A prior that is not a valid probability distribution is called an *improper prior*. A popular improper prior is the *flat prior* that has "density" identically equal to 1 over the entire range of permissible values.

Example: Model:

$$X_1, ..., X_{10} \text{ iid } N(0, \sigma^2).$$

Prior:

$$\sigma^2 \sim \text{ flat prior (1 on the entire positive real line).}$$

Posterior:

$$\sigma^2 \sim \text{ Inverse Gamma.}$$

♡

Exercise: Model:

$$X_1, ..., X_{10} \text{ iid } N(\mu, 1).$$

Prior:

$$\mu \sim \text{ flat prior}$$

Find the posterior of μ. ♡

6 Gibbs sampler in Bayesian problems

`http://192.168.50.201/statcomp/mcmc/gibbs3.html`

6.1 When and how to use a Gibbs sampler

To use Gibbs sampler to sample from a joint density $f(x_1, ..., x_k)$ we need to be able to simulate from all the full conditionals of the form

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

where $\mathbf{x}_{-i} = (x_1, ..., x_{i-1}, x_{i+1}, ..., x_k)$. So Gibbs sampler useful only where the joint density is complicated but all the full conditionals are easy to simulate from. In Bayesian statistics such joint densities occur frequently in the following situation.

Suppose we have a model with two parameters. For each parameter separately we have a conjugate prior *when the other parameter is known.* But if the both the parameters are unknown simulataneously (as is usually the case) then the posterior may become nasty. Gibbs sampler is applicable in such situations.

Example: Suppose that $X_1, ..., X_n$ are iid $N(\mu, \sigma^2)$. We know that *if σ^2 is known* then normal is a conjugate prior for μ. Similarly, *if μ is known* then inverse gamma is a conjugate prior for σ^2.

Now suppose that both μ and σ^2 are unknown and we assume the prior $\mu \sim$ normal $\sigma^2 \sim$ inverse gamma, and these are independent.

Now the posterior is a mess. However, since μ and σ^2 separately have conjugate priors, so the full conditionals are nice: the conditional posterior of μ given σ^2 is normal, and the other conditional is inverse gamma.

So we can easily implement the Gibbs sampler. ♡

6.2 Product conjugate prior situation

Let us generalise the above example. Suppose we have a model with k parameters $\theta_1, ..., \theta_k$. For each parameter we have a prior such that if the other parameters are known, then the posterior is nice (ie, we can simulate from it easily). Call this posterior as $POST_i$. Note that $POST_i$ is a probability distribution for θ_i. The distribution may (and, usually, does) involve the data as well as the other θ's. We assume that it is nice, ie, we can simulate θ_i from it (if we are given the data and the other θ's.) Conjugate prior is the most popular special case of this. Suppose that we want to simulate from the joint posterior of $\theta_1, ..., \theta_k$:

$$POST(\theta_1, ..., \theta_k),$$

which is usually nasty and often not available in a closed form. Observe that if $(\theta_1, ..., \theta_k)$is jointly distributed as $POST$, then the full conditionals are the $POST_i$'s.

Then we may use Gibbs sampler as follows to simulate from the joint posterior.

Gibbs sampler for product conjugate case

Start with some arbitrary values of $\theta_1, ..., \theta_k$.
For $i = 1, ..., n$
  For $j = 1, ..., k$
    Generate θ_j from $POST_j$
  End for
End for

6.3 A case study: Variance components with normal variables

Here we shall consider a random effects model with common variance.

- **Model:**

$$
\begin{aligned}
Y_{ij} &\sim N(\mu_i, \sigma^2) \ j = 1, ..., n_i \text{ and } i = 1, ..., p \\
\mu_i &\sim N(\theta, \eta^2) \ i = 1, ..., p
\end{aligned}
$$

- **Prior:**

$$
\begin{aligned}
\theta &\sim N(a, c^2) \\
\sigma^2 &\sim \text{ inverse gamma}(a_1, b_1) \\
\eta^2 &\sim \text{ inverse gamma}(a_2, b_2), \\
&\text{all indep.}
\end{aligned}
$$

Here a, c, a_1, b_1, a_2 and b_2 are all known numbers. θ, σ and η are the parameters. Note that μ's are *not* parameters.

First let us compute $POST_\theta$, ie the posterior of θ when σ and η are held fixed. Write down the soft part (ie the part involving θ) in

$$\text{prior of } \theta \times N(\theta, \eta^2) \times N(\mu_i, \sigma^2)$$

to see that it is of the form

$$exp(\text{some quadratic in } \theta)$$

So, by the inspection method, $POST_\theta$ is a normal distribution.

Similarly, $POST_{\sigma^2}$ and $POST_{\eta^2}$ are both inverse gamma.

Thus the Gibbs sampler takes the following form:

Start with some arbitrary values of $\theta, \sigma^2 and \eta^2$
For $i = 1, ..., n$
  Generate θ from $POST_\theta$ *(which is normal)*
  Generate σ^2 from $POST_{\sigma^2}$ *(which is inverse gamma)*
  Generate η^2 from $POST_{\eta^2}$ *(which is inverse gamma)*
End for

6.4 Data augmentation

Suppose that we apply Gibbs sampler to generate from $(X_1, ..., X_k)$. Let $X_i(t)$ be the value of X_i at time t. Then is the chain $\{X_i(t) : t = 0, 1, ...\}$ a Markov Chain? Not necessarily. This makes it somewhat difficult to analyse the converegnece of the individual components in a Gibbs sampler. However, if $k = 2$, then each component chain is a Markov chain in its own right. When $k = 2$, Gibbs sampler is also known as **data augmentation**. Though it is a special case of Gibbs sampler, it was proposed independently.

Suppose we want to simulate from (X, Y) where X, Y may be vector-valued. $X|Y$ has density f and $Y|X$ has density g. We assume that both f and g are known and easy to simulate from. Then the data augmentation algorithm starts with some arbitrary values for X and Y, and proceeds by alternately simulating X and Y.

7 Gibbs sampler in hierarchical Bayes

`http://192.168.50.201/statcomp/mcmc/gibbs4.html`

7.1 Hierarchical Bayes

A typical Bayes situation may be considered as a two stage random experiment. In the first stage nature generates the parameters from the priors, and the next she generates the data from the model using the parameters from the first stage.

Example: (Ordinary Bayes)

Data: X

Model: $X|\mu \sim N(\mu, 1)$.

Prior: $\mu \sim N(0, 1)$.

Here in the first stage we generate μ from $N(0, 1)$ and σ^2 from inverse gamma(2,3) independently of μ. Then in the second stage we generate X from $N(\mu, \sigma^2)$. ♡

In a hierachical Bayes problem the prior is itself specified in more than one stage.

Example: (Hierarchical Bayes) **Data:** X

Model: $X|\mu \sim N(\mu, 1)$

Prior: $\mu|\theta \sim N(\theta, 1), \theta|\eta \sim N(\eta, 1), \eta \sim N(0, 1)$.

Here the prior is given in 3 stages:

1. in the first stage we generate η from $N(0, 1)$,
2. then in the second stage θ from $N(\eta, 1)$,
3. then in the third stage μ using θ.

This finishes the parameter generation from the prior. Now in the final stage we generate X from $N(\mu, 1)$. ♡

Note that θ and η are not parameters of the model. They are parameters of the prior. Such "parameters" are called *hyperparameters*. Thus, here μ is a parameter and θ and η are hyperparameters.

The main aim of hierarchical Bayes, as of any Bayesian method, is to base the inference on the posteriors of the parameters. The posterior being often intractable, we want to use Gibbs sampling to simulate from it. In the above example we want to simulate from the posterior distribution of $\mu|X$. Note that we are interested only in the posterior of the parameter μ, and not of the hyperparameters.

It will be good at this point to compute the joint distribution of all the random variables involved. We shall need it later.

Example: (contd) The joint density of (η, θ, μ, X) is obtained by multiplying the prior densities and model density:

$$f(\eta, \theta, \mu, X) \propto exp(-1/2 * (\eta^2/2 + (\theta - \eta)^2/2 + (\mu - \theta)^2/2 + (X - \mu)^2/2)).$$

♡

7.2 Using Gibbs sampler for hierarchical Bayes problems

To use Gibbs sampler we need to do the following.

1. **List all the random variables involved.** Here we have 4 random variables:
$$\eta, \theta, \mu \text{ and } X.$$

2. **Draw the dependence graph.** This is a (directed) graph showing the stages mentioned above. In our example it is
$$\eta \to \theta \to \mu \to X.$$
This means that in order to generate X we need μ, to generate μ we need θ, and to generate θ we need η.

3. **List the full conditionals.** Recall that our aim was to simulate from the posterior of the parameter:
$$\mu | X.$$
Gibbs sampler for hierarchical Bayes does this by first simulating from the joint posterior of the parameters *and the hyperparameters*:
$$(\eta, \theta, \mu) | X,$$
and then throwing away the generated η and θ. So we need the following full-conditionals:
$$\eta | \theta, \mu, X$$
$$\theta | \eta, \mu, X$$
$$\mu | \eta, \theta, X.$$

4. **Simplify the full conditionals using the dependence graph.** Let us, for instance, try to compute the full conditional
$$\theta | \eta, \mu, X.$$
This is done by looking at the joint density and considering only the factor involving θ :
$$exp((\theta - \eta)^2/2 + (\mu - \theta)^2/2).$$
This is the soft part with respect to θ and is of the form exp(quadratic in θ). So the full conditional of θ is a normal distribution, $N(a, b)$, for some a, b.
To compute a and b, consider the density of $N(a, b)$:
$$f(\theta | \eta, \mu, X) \propto exp(-\theta^2/(2b) + (a/b)\theta)$$

Equating coefficients of θ^2 and θ here and in the soft part we see that $b = 2$ and $a = (\eta + \mu)/2$.

Note that $f(\theta|\eta, \mu, X)$ involves only μ and η, but not X. It was possible to infer it directly from the dependence graph, since μ and η are the neighbours of θ. This is a consequence of a general theorem given later in this lecture.

Similarly, $\mu|\eta, \theta, X \sim \mu|\theta, X$.

Exercise: What similar statement can you make about $\eta|\theta, \mu, X$? ♡

Exercise: Explicitly compute the full conditionals and check these statements. ♡

5. **Finally use the full conditionals to implement the Gibbs sampler.** In our example this amounts to:

Gibbs sampler for the hierarchical Bayes example

Start with some initial values: $\eta(0), \theta(0), \mu(0)$
For $i = 0, 1, 2, ..., N$
    Generate $\eta(i+1) \sim \eta|(\theta = \theta(i))$
    Generate $\theta(i+1) \sim \eta|(\eta = \eta(i+1), \mu = \mu(i))$
    Generate $\mu(i+1) \sim \eta|(\theta = \theta(i), X = \text{given data})$
End for
Output $\mu(N+1)$

7.3 Graphical models

In the above example the full conditionals could be simplified because of the dependence graph: the full condtionals involved *only* the neighbouring nodes. This concept can be generalised to produce *graphical models.*

Example: (Graphical model) Rocke-Lorenzato (RL) model: In chemistry one often wants to relate some measurement, Y, (e.g., electrical conductivity) to the known concentration, x, of a chemical solution. For this the measurement is taken J times for each of I known concentration levels, x_i. Let the j-th measurement for the i-th concentration level be Y_{ij}. Then the RL model relates Y_{ij}'s to the x_i's as follows.

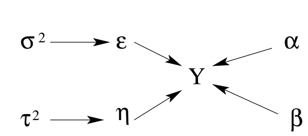
$$Y_{ij} = \alpha + \beta x_i exp(\eta_{ij}) + \epsilon_{ij},$$

where η_{ij}'s are iid $N(0, \sigma^2)$ and ϵ_{ij}'s are iid $N(0, \tau^2)$, and they are independent.

So we have four parameters

$$\alpha, \beta, \sigma^2, \tau^2.$$

We use flat priors for all the four parameters. This produces the following graphical model.



Here we have 7 random variables in this model and these correspond to the 7 nodes: $Y, \alpha, \beta, \eta, \epsilon, \sigma^2$ and τ^2. ♡

Defn

> Suppose that we have a graphical model. Let X and Y be two nodes in it. We say that X *affects* Y if any one of the following is true:
>
> 1. $X \to Y$
> 2. $Y \to X$
> 3. there is a node Z such that $X \to Z$ and also $Y \to Z$.
>
> This is not a standard definition. It is for use only inside this course.

Example: (RL model, contd.) Here σ^2 and τ^2 do not affect α or β, and vice versa. However, all the other pairs affect each other. ♡

Here is an important theorem for graphical models, which has a direct proof.

Theorem

> In a graphial model the conditional distribution of any node given *all the rest* is the same as the conditional distribution of the node given *only* the nodes that affect it.

Example: (RL model, contd.) By the theorem above, full conditionals of α and β do not involve σ^2 and τ^2. The full conditional of σ^2 involves only η. Similarly, that of τ^2 involves only ϵ. And so on. ♡

In general, the entire graph may be huge and the model very complicated, but the number of nodes affecting a given node is often small. This keeps the full conditional simple, and makes graphical models an ideal candidate for the Gibbs sampler.

8 Adaptive rejection sampling (ARS)

`http://192.168.50.201/statcomp/mcmc/gibbs5.html`

8.1 Introduction

Gibbs sampler requires one to simulate from all the full conditionals. Sometimes, however, a full conditional may be complicated. Then Gibbs sampler cannot be applied directly.

Example: Suppose that we have data X.

Model: $X|p, \lambda \sim Gamma(p, \lambda)$

Prior: $p \sim Expo(1) and \lambda \sim Expo(3)$ indep.

Here the joint density of (p, λ, X) is

$$f(p, \lambda, X) \propto exp(-p)exp(-3\lambda)\lambda^p X^{p-1} exp(-\lambda X)/GAMMA(p)$$

Collecting terms involving λ, we get the full conditional density of λ as

$$f(\lambda|p, X) = (\text{constant free of } \lambda) \times \lambda^p exp(-(X+3)\lambda)$$

By inspection we recognise the full conditional as Gamma$(p, X+3)$. Similarly, collecting the terms involving p, the full conditional density of p is

$$f(p|\lambda, X) = (\text{constant free of } p) \times exp(-p)\lambda^p X^{p-1}/GAMMA(p),$$

which is not a standard distribution. It is not obvious how we may simulate from this full conditional. ♡

In such a situation we may try to simulate from the few tough conditionals using a form of rejection sampling, called Adaptive Rejection Sampling. For this to work we need the conditional to be log-concave. But first let us review the concept of rejection sampling in general.

8.2 Rejection sampling

Suppose that we have a complicated density f that we *want* to sample from. Suppose also that we have a density g such that

$$f(x) < Bg(x) \text{ for all } x,$$

where B is some known constant. We shall call Bg an *envelope* of f. Sometimes we shall abuse notation and call g alone as an envelope without explicitly mentioning the constant B. Assume that we can simulate from this density, g.

Then rejection sampling is a method to modify iid samples from g to get iid samples from f as follows.

First generate $X \sim g$. Compute the ratio

$$R = f(X)/(Bg(X)).$$

Note that since X is generated from g, the chance that the denominator vanishes is 0. Hence R is well-defined with probability 1. Toss a coin with probability of head equal to R. If you get head, return X, else repeat the process. Here is the algorithm.

Rejection sampling

LOOP: Generate $X \sim g$.
Let $R = f(X)/(Bg(X))$.
Toss a coin with $P(head) = R$
If head
  Output X.
Else
  Go to LOOP.
  End if

Exercise: Can you prove that this algorithm really results in a rv with the correct density? [Hint: Define a 0-1 rv Y which is 1 iff X is accepted. You know that $X \sim g$. Find $Y|X$. Then compute the joint distribution of (X, Y). You are to show that $X|Y = 1 \sim f$. You need to use nothing but elementary probability.] ♡

Exercise: Show that this algorithm terminates with probability 1. ♡

You'll be well advised to try out some simulation now with this scheme.

MATLAB

```
function X = rej(f,B)
  while(1)
    X = unifrnd(0,1); % g is Unif[0,1] density
    numerator = feval(f,X);
    denominator = B; % g(X) = 1 since g is Unif[0,1] density

    R = numerator/denominator;

    u = unifrnd(0,1);
    if(u < R)
      return;
    end
  end
```

Here is a sample density

MATLAB

```
function d = tent(x)
```

```
if x < 0.5
  d = 4 x;
else
  d = 4- 4 x;
end
```

8.3 Adaptive rejection sampling (ARS)

The main step behind any rejection sampling is to find a tight envelop. It usually requires some computation to find one. This extra computation is well worth the effort if we are going to draw a large number of samples from the distribution. But in Gibbs sampling we draw only one random sample at each step, and the distribution changes from step to step. So using rejection sampling in the usual way is a wastage in a Gibbs sampler. The following example illustrates this point.

Example: Here is the Gibbs sampler to simulate from the joint distribution of (X, Y):

Start with some $X(0), Y(0)$
For $i = 0, 1, ...N$
  Generate $X(i+1) \sim X|Y = Y(i)$
  Generate $Y(i+1) \sim Y|X = X(i+1)$
End for

Clearly, the full conditionals are changing from step to step. For instance, $X(1)$ is generated from $X|Y = Y(0)$ while $X(2)$ is generated from $X|Y = Y(1)$. These two conditional distributions may not be the same. ♡

However, if the target density is log-concave *i.e.,*, its log is a concave function), then we can use an adaptive version of rejection sampling called Adaptive Rejection Sampling or ARS. The main idea in ARS is to carry out rejection sampling according to the usual method. But each time we reject a point, ARS tries to use the rejected point to tighten the envelope. This reduces the chance of rejection in future. Here is the basic structure of the algorithm. Note that it is the same as the rejection sampling algorithm except for the underlined line.

Adaptive Rejection Sampling (basic structure)

LOOP: Generate $X \sim g$.
Let $R = f(X)/(Bg(X))$.
Toss a coin with $P(head) = R$
If head
  Output X.
Else
  <u>Update the envelope using X.</u>
  Go to LOOP.
End if

8.4 A general class of envelopes

In order to tighten the envelope at each step, ARS uses a special class of envelopes that we discuss now. First, a theorem about the general shape of any log-concave density, which makes ARS possible.

Theorem

A log-concave density is always unimodal.

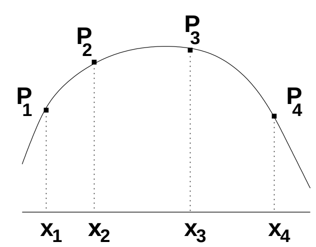
Thus, for a typical log-concave density f, the graph of $log(f)$ looks like the curve in the following diagram.

Take a finite set

$$S = \{x_1 < ... < x_k\}$$

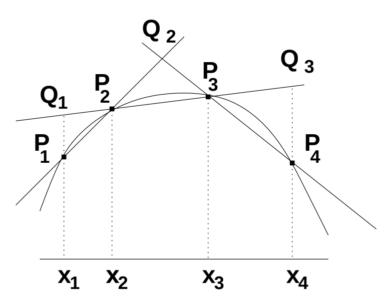
of k points ($k > 2$), on the horizontal axis, so that the line joining $(x_1, log(f(x_1)))$ and $(x_2, log(f(x_2)))$ should has a positive slope and the line joining the last two points has negative slope. Given any such set S we shall construct an envelope G_s for $log(f)$ as follows. (Then $g_s := exp(G_s)$ will be an envelope for f.)

Draw vertical lines through each x_i to intersect the $log(f)$-curve at P_i.



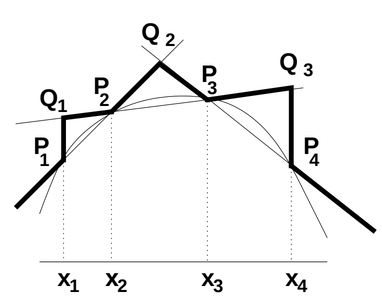
Graph of $log(f)$

For $i = 1, ..., k-1$ join P_i and P_{i+1} using straight lines.



Join with lines

Now consider the zigzag crown shown below.



Zigzag crown shown with thick line

Note that this zigzag crown is an envelope over $log(f)$. The required envelope g_s over f is the density proportional to its exponential. Thus, g_s is a piecewise exponential curve.

8.5 Tightening the envelope

Note that a larger S produces a tighter envelope. The main idea behind ARS is to enlarge S at each rejection.

Here is the algorithm (Ref: Appl Stat (1992, 41) Gilks):

Adaptive Rejection Sampling (complete)

Initialise S
LOOP: Generate $X \sim g_S$.
Let $R = f(X)/(B_s g_s(X))$.
Toss a coin with $P(head) = R$
If head
  Output X.
Else
  Add X **to** S.
  Go to LOOP.
End if

Usually at most 5 or 6 iterations are required before the algorithm outputs a number.

8.6 Initialising S

At the very beginning we start with S of size 3:

$$S = \{x_1, x_2, x_3\},$$

where the line joining $(x_1, log(f(x_1)))$ and $(x_2, log(f(x_2)))$ has a positive slope and the line joining $(x_2, log(f(x_2)))$ and $(x_3, log(f(x_3)))$ has a negative slope.

During Gibbs sampling the full conditionals change only slightly from one iteration to the next. So the S from the end of one iteration provides a good starting S for the next iteration.

Exercise: Write a Matlab function or SAS module to simulate from the density proportional to $exp(ax + b)$ over (l, r) where l may be $-\infty$ or r may be ∞. In these unbounded cases, a must be negative. ♡

Exercise: Given an S write down a ready-to-implement algorithm to simulate from g_s. ♡

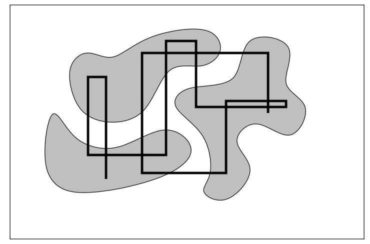
9 Reparametrisation to speed up convergence

`http://192.168.50.201/statcomp/mcmc/.html`

9.1 Reparametrisation

Gibbs sampling is a local sampling procedure in the sense that it makes steps whose distributions are determined by the current conditionals. It would be a good idea to acquaint ourselves with the notion of current conditionals before proceeding further. Throughout this page we shall use bivariate examples to help ourselves visualize the conditionals. However, the principle remains equally valid in higher dimensions.

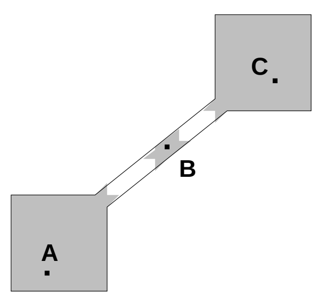
Example: Suppose we are to simulate from the bivariate density $f(x, y)$. Gibbs sampler proceeds by making alternate X-steps and Y-steps. So the "path" of the Gibbs sampler is like a zigzag line with alternate horizontal and vertical segments.



All the points are contained in the support of $f(x, y)$ (shown shaded above), though parts of the line segments may fall outside the support, if the support is not convex. Suppose that currently we are at a point (a, b) and we are about to make an Y-step. This step is made by simulating from the conditional distribution $Y|X = a$. Pictorially, this corresponds to moving along the vertical line according to the density proportional to $f(a, y)$. ♡

Ideally, we want a Gibbs sampler to mix quickly, *i.e.,* make large steps and to keep on visiting all the nooks and corners of the space. However, if a Gibbs sampler once gets into a narrow region from where not much of the entire space is visible, it often gets "stuck" in that narrow region, *i.e.,* keep on making only small steps inside the "visible" region.

Example: Suppose we want to simulate from the bivariate uniform distribution over the region shown below.



It is a single connected region made of two large chambers connected by a narrow oblique corridor (a bottleneck). A, B and C are three typical points in this region. The shaded area surroundng each point shows the part of the region "visible" from that point, i.e. reachable by one pair of X- and Y-step. When the current point is in one of the chambers (e.g., A or C) the other chamber is not "visible" to it. Only the entrance to the bottleneck is visible. So it continues for a long time in the same chamber and then enters the bottleneck. Once inside the bottleneck (e.g., B) hardly anything else is visible, so the chain keeps on making small steps inside, until it accidentally slips out into one of the two chambers. Then it stays long in it and so on.

This behaviour is an example of *slow mixing.*

The main problem here is that the chambers are badly connected in the horizontal/vertical direction. These directions are important in a Gibbs sampler because it can move in these two directions only. Reparametrisation often solves this problem. In this case, we can just rotate the sample space until the chambers are on the same vertical or horizontal level. Now we do not need the bottleneck to move from one chamber to the other. ♡

The above example shows an example of reparametrisation. We moved from (X, Y) to (U, V), where (U, V) is the rotated version. Then we performed Gibbs sampling, and finally rotated back to (X, Y). It may seem unusual that this is called "reparametrisation" and not something like "transformation". This is because Gibbs sampling is most commonly used to simulate from posteriors. There X and Y are actually parameters, so transforming them amounts to reparametrising the original problem.

Example: Suppose we want to generate data from bivariate normal using Gibbs sampling. Let

$$(X, Y) \sim N([\mu_1 \mu_2], [\sigma_{11}\sigma_{12}; \sigma_{21}\sigma_{22}])$$

Then the full conditionals are

$$X|Y \sim N(\mu_1 + (\sigma_{12}/\sigma_{22}) * (Y - \mu_2), \sigma_{11} - \sigma_{12}^2/\sigma_{22})$$

and

$$Y|X \sim N(\mu_2 + (\sigma_{12}/\sigma_{11}) * (X - \mu_1), \sigma_{22} - \sigma_{12}^2/\sigma_{11})$$

If the covariace matrix Σ is near singular then the bivariate normal distribution is tightly spread along a line. In this case each X-step and Y-step will tend to be small. However, if we scale and rotate (X, Y) then the resulting distribution will be more spherical and the mixing would be better. ♡

9.2 How to detect the problem

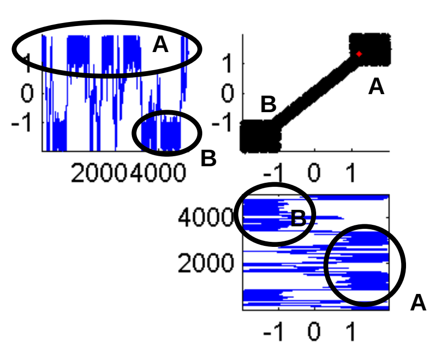
In the examples above we deliberately took simple target distributions in the plane so that we can explicitly see if the chain is mixing slowly. However, in practice this is seldom possible. There we use two methods to detect possible slow mixing.

- Time series plot
- Correlation computation

9.2.1 Time series plot

Look at the time series plot of the generated sequences. The plot should look homogeneous. Existence of stable levels connected by abrupt jumps is a sign of slow mixing.

Example: (Bottleneck example, contd) Here is a sample run from the bottleneck example above.



The top right hand plot is the scatterplot of the generated points. The other two show the time series plots of the X's and the Y's. Note the existence of the stable levels (shown as A and B). The regions marked A, for instance, correspond to the chain being stuck in the upper chamber. B denotes when it is stuck inside the other chamber. The abrupt jumps connecting the two levels denote accidental slips through the bottleneck. ♡

9.2.2 Correlation computation

Another cause is high correlation among the generated sequences. In such situations it is better to reparametrise to get rid of this correlation. The bivariate normal example above illustrates this situation. It is often a good idea to estimate the covariance matrix of (X, Y) and transform (X, Y) appropriately to make the covariance matrix as close to identity as possible.

9.3 Mixture distribution

If the target distribution is the mixture of two or more distributions then we can consider the situation as two large chambers connected via a narrow bottleneck. The Gibbs sampler is like a fly hopping at random in this space. It will spend a long time in th same chamber before accidentally moving to the other, then it will spend a long time in that chamber before returning. This is a frequently occuring case of slow mixing. Reparametrization is often the cure.

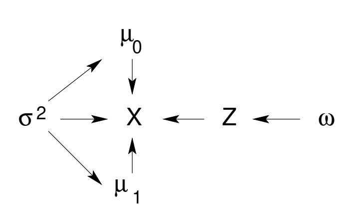
Example: Consider the following graphical model.

Model : We have two normal distributions $N(\mu_0, \sigma^2)$ and $N(\mu_1, \sigma^2)$. Also we have a $n \times 1$ vector Z where each entry is either 0 or 1. Here the parameters are μ_0, μ_1, σ^2 and Z. Our data set is $X_1, ..., X_n$, where X_i follows $N(\mu_k, \sigma^2)$ if $Z_i = k$.

Prior :

$$\begin{aligned}
Z_1, ..., Z_n \text{ iid } Bern(w). \\
\text{[Let } n_1 \text{ of these be } 0's, \text{ and } n_2 \text{ be 1's.]} \\
w \sim Beta \\
\sigma^2 \sim \text{ inverse gamma} \\
\mu_1 | \sigma^2 \sim N(a_1, \sigma^2/n_1) \\
\mu_2 | \sigma^2 \sim N(a_2, \sigma^2/n_2)
\end{aligned}$$

Note that we have the following graphical model situation:



Full conditionals :

$$\begin{aligned}
\sigma^2 | rest \sim \text{ inverse gamma} \\
\mu_1 | rest \sim normal \\
\mu_2 | rest \sim normal \\
w | rest \sim w | Z \sim Beta
\end{aligned}$$

This is easy to implement. But if the two component normals are far apart then mixing becomes slow. But if we reparametrise μ_1 to $(\mu_0 + \theta)$ then the problem goes away. ♡

10 Metropolis-Hastings algorithm

`http://192.168.50.201/statcomp/mcmc/mh1.html`

10.1 Introduction

This is a scheme for MCMC which is quite general and subsumes Gibbs sampling as a special case. In Gibbs sampling we need to be able to simulate from the all the full conditionals. No such condition is required for the Metropolis-Hastings algorithm. In spirit it is quite similar to rejection method of sampling. There we sampled from an envelope density and then filtered the sampled point through a coin-toss stage. The P(head) for the coin was so chosen that the final output followed the target density. We can call the initial sampling from the envelope as "proposal", which is accepted/rejected by the coin toss. In a Metropolis-Hastings algorithm we similarly filter a Markov chain, as we discuss below.

10.2 Filtering a Markov chain

A Markov chain moves by making random steps. If it is at $X(t)$ at time t then it generates Y based on the transition kernel, and moves there: $X(t+1) = Y$. To filter the Markov chain we toss a coin with some probability of head that may depend on t as well as $X(t)$. If we get a head, then we move to Y as before: $X(t+1) = Y$. But if we get a tail then we stay where we are: $X(t+1) = X(t)$.

Filtering a Markov chain

Input: A Markov chain with state space S, an initial state $X(0)$ and a filter probability function $H : S \times S \to [0, 1]$.
Output: A realization of the filtered Markov chain.
For $t = 0, 1, 2, ...$
  Generate Y starting the given Markov chain from $X(t)$
  Toss a coin with $P(head) = H(X(t), Y)$ independently of the last step.
  If head
    Let $X(t+1) = Y$
  Else
    Let $X(t+1) = X(t)$
  End if
  Output $X(t+1)$
End for

Example: Consider the Markov chain with state space

$$S = \{1, 2, 3\},$$

and transition matrix

$$P = \begin{bmatrix} 0.0 & 0.5 & 0.5 \\ 0.5 & 0.0 & 0.5 \\ 0.5 & 0.5 & 0.0 \end{bmatrix}$$

You can visualise the states as three vertices of a triangle. At each step we move to either of the other two vertices with probability 0.5 each.

It is easy to see that this Markov chain is irreducible and aperiodic. So it converges to a unique limiting distribution irrespective of initial distribution. By inspection we see that the uniform distribution is a stationary distribution of the chain. So it must the the unique limiting distribution. We shall use this Markov chain as our proposal chain. For the accept/reject stage we shall use the following matrix of probabilities:

$$\begin{bmatrix} 0.5 & 0.3 & 0.4 \\ 0.5 & 1.0 & 0.0 \\ 0.2 & 1.5 & 1.0 \end{bmatrix}$$

We shall call this matrix as R. Here is how to filter the proposal chain using R.

We start at $X(0) = 1$, say. Then we run one step of the proposal chain. It produces $Y = 1$ or 2 with probability 0.5 each. Suppose, we get $Y = 3$. Thus we are at 1, and propose to move to 3. Look up $R(1, 3) = 0.4$. We toss a coin with $P(Head) = 0.4$.If we get Head then we accept the proposal, i.e., set $X(1) = Y = 3$, otherwise we stay where we are, i.e., set $X(1) = X(0) = 1$. We do the same thing at each step.

Note that the filtered Markov chain has the following transition matrix:

$$\begin{bmatrix} * & 0.15 & 0.20 \\ 0.25 & * & 0.00 \\ 0.10 & 0.50 & * \end{bmatrix}$$

Here each * is computed to keep the row sums equal to 1.

♡

10.3 Properties of the filtered chain

Henceforth we shall call the original chain as the **proposal chain**. Note that the filtered transition matrix is quite different from the proposal chain, and may have completely different convergence properties.

First notice that the filtered chain is again a Markov chain, since the filtering merely changes the randomization mechanism of each step without using the past history of the chain. The following result is easy to prove.

Theorem

> If the proposal chain is irreducible and $0 < r(x, y) < 1$ for each x, y, then the filtered chain is irreducible and aperiodic.

The Metropolis-Hastings algorithm tries to design a suitable R such that the filtered Markov chain converges to the required target density.

Example: Again consider the same proposal Markov chain as in the last example. We want to design R in a way that the filtered Markov chain will converge to the distribution q where

$$q(1) = 0.5, q(2) = 0.25, q(3) = 0.25.$$

Here is one way. Suppose we are at i and the proposal chain has proposed a move to j. Then take

$$R(i, j) = min\{1, q(j)/q(i)\}.$$

Thus we surely accept the move it $q(j) > q(i)$. Otherwise, we accept with probability $q(j)/q(i)$.

Intuitively, we are eager to move to states with high q-probability.

It may not be readily obvious why the filtere chain should converge to q. This will follow from the theorem about Metropolis algorithm to be presented shortly. ♡

Exercise: Write down the transition kernel of the filtered chain. ♡

10.4 Metropolis-Hastings algorithm

In a Metropolis-Hastings algorithm we have a **proposal Markov chain**. This Markov chain need not converge to the target density. At each step we filter the output of the proposal Markov chain, so that the filtered Markov chain converges to the required target density.

The above example is a special case of the Metropolis-Hastings algorithm. Here is the general algorithm.

Set up: We have a proposal Markov chain with transition kernel $k_x(y)$. (Thus, if we are at $X(i) = x$, then we propose to move to $Y \sim k_x(y)$.) We have a target density $q(x)$ that we want to sample from. Our aim is to filter the proposal Markov chain so that the filtered Markov chain converges to q.

Metropolis-Hastings algorithm

Input: A target distribution q on a space S.
[It is enough to know q up to an unknown normalizing constant.]
An irreducible proposal chain with state space S.
Output: A Markov chain on S converging to q.
Start with any $X(0)$ in S.
For $t = 0, 1, ...$
  Generate Y by running the proposal chain from $X(0)$.
  Let $r = (q(y)k_y(x))/(q(x)k_x(y))$.
  Toss a coin with $P(head) = min\{1, r\}$.
  If head
    X(t+1) = Y
  Else
    X(t+1) = X(t)
  End if
End for

In practice, the target density q is given to you. So designing a Metropolis-Hastings algorithm means choosing the proposal chain. (Once that is done, the acceptance probability is automatically determined by it and q.) Choosing a proposal Markov chain is usually easy, since it does not need to satisfy too many constraints. In fact, the conditions mentioned in the following theorem suffice.

Theorem

If $k_x(y) > 0$for all x, y in the support of q, and if there is positive holding probability, then the Metropolis Hastings chain converges to q irrespective of initial condition.

Holding probability is the chance that the Markov chain does not move at some step. Thus, the holding probability at step i is $P(X(i) = X(i+1))$. In terms of the Metropolis-Hastings chain, it is basically the probability of a tail in the accept/reject coin toss.

Example: Consider the following problem.

Model:

$$X_1, ..., X_n \text{ iid } N(0, \sigma^2).$$

Prior:

$$\sigma \sim Expo(1).$$

Then the posterior density of σ is proportional to

$$exp(-\sigma)exp(-SS/(2\sigma^2))/\sigma^n,$$

where SS denotes the sum of squares of the X_i's. The normalizing constant is quite difficult to compute. But that is no problem at all for the Metropolis-Hastings algorithm. Let us take a very simple proposal chain, Q: it is just an iid $Expo(1)$ sequence! Thus

$$q(x,y) = \begin{array}{ll} exp(-y) & \text{if } y > 0 \\ 0 & \text{else} \end{array}$$

Note that it is free of x! ♡

10.5 Using the computer

We shall do the above computation with $n = 10, SS = 15.46$:

MATLAB

```
function newSigma = mh(oldSigma)
  newSigma = exprnd(1);
  num = exp(-15.46/(2*newSigma^2))/newSigma^10;
  deno = exp(-15.46/(2*oldSigma^2))/oldSigma^10;

  R = num/deno;

  if unifrnd(0,1) < R
     return
  end

  newSigma = oldSigma;
```

You may use this from the following script:

MATLAB

```
sigma = ones(100,1000);
for i = 1:100
  for j = 1:999
    sigma(i,j+1) = mh(sigma(i,j));
  end
end
```

Not in syllabus

First let us work out the kernel of a MH chain. Since the chain allows *holding* (ie, staying where you are) the kernel is in general a mixture of density and mass. Check that it is

$$M_x(y) = R(x,y)k(x,y) + (1 - R(x))\delta_x(y),$$

where

$$R(x) = \int r(x,y)k(x,y)dy,$$

and δ_x is unit mass at x.

Exercise: Show this. ♡

Theorem

If the support of $k_x(.)$ contains the support of q for all x, then q is a stationary distribution of the MH chain.

Proof:

1. Let $M_x(y)$ as above denote the transition kernel of the MH chain. We shall show first that it satisfies the *detailed balance condition* wrt q, which means:
$$q(x)M_x(y) = q(y)M_y(x).$$
[Intuitively, if we start the chain with initial distribution f, then the lhs is the likelihood of starting at x and moving to y in the first step. The rhs is the likelihood of starting at y and moving to x in the first step.]
This is shown easily by proving that
$$q(x)R(x,y)M_x(y) = q(y)R(y,x)M_y(x)$$
and
$$q(x)(1 - R(x))\delta_x(y) = q(y)(1 - R(y))\delta_y(x)$$

2. Here we shall show that if $M_x(y)$ satisfies the detailed balance condition wrt q, then q must be a stationary distribution. Thus, if we start the chain with $X_0 \sim q$, then $X_1 \sim q$, as well. Take any measurable set B. We already know
$$P(X_0 \text{ in } B) = \int_B (q(x)dx).$$
Now
$$\begin{aligned} P(X_1 in B) &= \int_S q(y)M_y(B)dy \\ &= \int_S q(y) \int_B M_y(x)dxdy \\ &= \int_B \int_S k_x(y)q(x)dydx \\ &= \int_B q(x)dx \end{aligned}$$

♡

Theorem

If $k(x, y) > 0$ for all x, y in the support of f then the MH chain is irreducible.

Theorem

If a MH chain is irreducible then it is Harris recurrent.

Theorem

An irreducible MH chain is aperiodic if there is strictly positive holding probability.

Theorem

For an irreducible aperiodic MH chain, we have the following:

- For any L_1 function h,

$$\sum h(X_i)/N \to E(h(X)) a.e.[q].$$

-

$$X_n \to q \text{ in distribution.}$$

Here is a less restrictive sufficient condition for irreducibility and aperiodicity of MH chains.

Theorem

Suppose that the support of f is connected. If is positive and bounded on every compact subset of its support, and if there exist positive ϵ, δ such that

$$k(x,y) > \epsilon \text{ whenever } |x-y| < \delta,$$

then the chain is aperiodic and irreducible.

11 Different types of Metropolis-Hastings algorithm

`http://192.168.50.201/statcomp/mcmc/mh2.html`

The MH scheme is very general, some special cases being more popular than others. Here we shall take a look at some of the them.

11.1 Gibbs sampler as MH

Theorem

Gibbs sampling is a special case of the MH algorithm.

Proof:

Let us consider the 2-dim case where we want to simulate from the joint density of some (X, Y). The Gibbs sampler proceeds by making alternate X-steps and Y-steps. Suppose that we are currently at (a, b) and are about to make an X-step. Let $X|Y = y$ have density $q(x|y)$. Consider the proposal kernel

$$\begin{aligned} k_{(a,b)}(x,y) &= q(x|b) \text{ if } y = b \\ &= 0 \text{ else} \end{aligned}$$

The condition "$y = b$" keeps the proposal confined to the horizontal line through (a, b). Now let us compute $r_{(a,b)}(x, y)$ for this proposal. It is

$$(q(x,y)k_{(x,y)}(a,b))/(q(a,b)k_{(a,b)}(x,y))$$

The numerator is

$$\begin{aligned} q(x,y)k_{(a,b)}(x,y) &= q(x,y)q(x|b) \text{ if } y = b \\ & \quad 0 \text{ else} \end{aligned}$$

Similarly, the denominator is

$$\begin{aligned} q(a,b)k_{(x,y)}(a,b) &= q(a,b)q(a|y) \text{ if } b=y \\ & \quad 0 \text{ else} \end{aligned}$$

Let $q(y)$ denote the marginal of y. Then

$$q(x|y) = q(x,y)/q(y).$$

Using this show that the numerator and denominator or $r_{(a,b)}(x,y)$ are actually the same. So $r_{(a,b)}(x,y) = 1$.

Similar argument holds for the Y-step.

Thus the Metropolis-Hastings algorithm with this proposal chain is the sme as the Gibbs sampler. ♡

Example: Let us go through the steps of the above proof using

$$\begin{aligned} q(x,y) &= exp(-2y)y^x/x! \text{ if } y>0 \text{ and } x=0,1,... \\ & \quad 0 \text{ else.} \end{aligned}$$

By inspection it is easy (how?) to see that $X|Y\ Poi(Y)$, i.e.,

$$q(x|y) = exp(-y)y^x/x! \text{ if } x=0,1,...$$

So the proposal kernel for the X-step is

$$\begin{aligned} k_{(a,b)}(x,y) &= exp(-b)b^x/x! \text{ if } x=0,1,... \text{ and } y=b \\ & \quad 0 \text{ else.} \end{aligned}$$

Thus, the numerator of $r_{(a,b)}(x,y)$ is

$$\begin{aligned} q(x,y)k_{(a,b)}(x,y) &= exp(-3y)y^{a+x}/(a!x!) \text{ if } y=b>0 \text{ and } a,x=0,1,... \\ & \quad 0 \text{ else} \end{aligned}$$

The denominator is

$$\begin{aligned} q(a,b)k_{(x,y)}(a,b) &= exp(-3b)y^{x+a}/(x!a!) \text{ if } b=y>0 \text{ and } x,a=0,1,... \\ & \quad 0 \text{ else} \end{aligned}$$

Clearly these are the same. ♡

Exercise:

- Do similar, explicit computation for the Y-step in the above example.
- Where do you think the joint density $q(x,y)$ in the above example will occur in Bayesian statistics?

♡

11.2 Independent MH algorithm

Here the proposal density does not depend on the current point. Thus,

$$k_x(y) \text{ is free of } x.$$

Example: (Gamma generation) Generating Gamma(p, λ) is easy when p is an integer, because it is sum of p iid Expo(λ) random variables. To simulate from Gamma(p, λ) when p is is not an integer we may use independent MH algorithm with proposal density Gamma$(ceil(p), \lambda)$, where $ceil(p)$ is the smallest integer exceeding p.

♡

Let the common proposal density be $f(x)$. Then we have the following theorem.

Theorem

If

$$q(x) \leq M f(x) \text{ for all } x \text{ in support of } q,$$

for some constant M, then

$$||p_n - q||_{TV} \leq 2(1 - 1/M)^n$$

11.3 Random Walk MH algorithm

Here the family of proposal densities is constructed by translating some common underlying density f. Thus,

$$k_x(y) = f(y - x).$$

It is called random walk MH because, if we look at only the proposals (ie, if we always accept the proposals) then we have a random walk. Notice that here

$$k_x(y) = k_y(x).$$

So $R(x, y)$ is free of k.

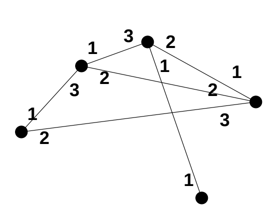
This method may be nice if we already have a random walk that explores well the support of the target distribution.

11.4 Metropolis-Hastings on finite graphs

Suppose that we want to simulate from the distribution on a finite (but possibly HUGE) set S. We are given a connected graph with S as its vertex set. Self-loops or multiple edges between same pair of vertices are not allowed. Assume that we know a number d such that the degree of each vertex is at most d, with at least one vertex having degree strictly less than d.

Exercise: Show that if $|S| = n$, then we can always take $d = n + 1$. However, in practice, we can often find a suitable d much smaller than $|S|$. ♡

Let the edges incident at each vertex be labelled with numbers from $\{1, 2., ...d\}$.



Example: For the above graph we have taken $d = 3$. ♡

Now consider the following Markov chain on this graph. At each step we pick up a number X uniformly from {1,...,d}. If there is an X-th edge from the current vertex, then we follow it, else we stay at the current vertex.

Exercise: Compute the transition matrix for the graph from the last example. ♡

Note that the transition matrix is symmetric. So uniform distribution is a stationary distribution.

Exercise: Show that for any Markov chain with a symmetric transition matrix, uniform distribution is a stationary distribution. ♡

Since the graph is connected, the Markov chain is irreducible. Since there is at least one vertex with degree less than d, the Markov chain is aperiodic. So the Markov chain always converges to uniform distribution.

Suppose next that we have a probability q on S. We want to simulate from q. Then we can apply the above Markov chain as the proposal in the Metropolis-Hastings algorithm.

The proposal chain being symmetric, the acceptance probability is

$$min\{q(y)/q(x), 1\}$$

for moving from x to y. Note that q enters into the picture only via the ratio above, so it is enough to know q upto a constant factor.

This method of designing a Metropolis-Hastings algorithm uses this idea to simulate distributions over large finite sets, S. It constructs a low-degree graph with S as the vertex set, and then performs the above algorithm.

11.5 Hit and run algorithms

This is actually a speacial case of random walk MH. Here we pick a random direction in space and move by a random amount. This may be a good method to avoid bottlenecks.

11.6 Adaptive Rejection Metropolis Sampling

This is a fully automated version of MH algorithm, where you only need to specify the target density (possibly upto a constant factor), and ARMS chooses appropriate proposal densities for you. This method is a generalisation of ARS for Gibbs sampler. However, unlike ARS, it works even if the target density is not log-concave.

Suppose that the target density is proportional to q, which is our input. Let $h(x) = log(q(x))$. We shall approximate $h(x)$ by a piecewise linear function g(x), and then use $min\{q(x), exp(g(x))\}$ as a proposal density (upto a constant factor.)

To get $g(x)$ we pick a set B consisting of $k+2$ numbers on the x-axis. Let g be the zigzag crown corresponding to these points.

To simulate from the proposal density which is proportional to

$$s(x) = min\{q(x), exp(g(x))\}$$

we use usual accept/reject sampling using exp(g(x)) as the envelope. For every rejection we add the rejected point to B, thus improving the chance of acceptance in the next stage. Here is the entire algorithm.

Adaptive Rejection Metropolis Sampling (ARMS)

Start with q, a small B, and $X(0)$.
Let t = 0.
ENV: Compute the function exp(g(.)).
Generate Y from density proportional to this function.
Toss a coin with $P(head) = q(Y)/exp(g(Y))$.
If head
  Add Y to B.
  Go to ENV.
End if
Toss a coin with $P(head) = (q(Y)s(X(t)))/(q(X(t))s(Y))$
If head
  X(t+1) = Y
Else
  X(t+1) = X(t)
End if

12 Convergence issues

`http://192.168.50.201/statcomp/mcmc/conv.html`

12.1 Introduction

Any MCMC scheme is an iterative one that needs to be run upto convergence. If $X(k)$ denotes the random iterate at step k, then this convergence is in terms of the *distribution* of $X(k)$. If $p(k)$ denotes the distribution of $X(k)$, then we need $p(k)$'s to come closer and closer. This does not mean that the $X(k)$'s will come closer together. Since we have only realisation, viz, $X(k)$ from $p(k)$, this checking is complicated. Here we shall present a number of ad hoc methods that consider $\{X(k)\}$ as a time series and test for its stationarity.

12.2 Plotting

Never forget to plot the the $X(k)$'s against k. Plots often reveal valueable information about lack of convergence. Clustring of $X(k)$'s in one region before jumping to another region is one indication of lack of convergnece. Also, plotting the components against one another may reveal correlations responsible for slow mixing.

12.3 Gelman and Rubin's test

Run m chains independently in parallel: $\{X_i(k)\}$. Test for equality of means. Let F denote the usual F-statistic. Then Gelman and Rubin suggests using

$$R = (n-1)/n + F/n,$$

where n is the length of the chain. R "far" from 1 indicates lack of convergence.

12.4 Gewecke's test

Gewecke's idea is to carry out a similar test for equality of means. However, instead of comparing between parallely run chains, he suggests comparing two parts of the same chain. Pick up two regions in the chain: A and B. A is of size n_A and B of n_B. Now perform a t-like test:

$$Z = (\bar{X}_A - \bar{X}_A)/sqrt(s_A/n_A + s_B/n_B),$$

where $s_A$and s_B are suitable variance estimates using spectral analysis. If Z is "far" from 0, then there is reason to worry.

12.5 Dickey-Fuller test

The test statistic here is

$$n(\hat{\gamma}(1)/\hat{\gamma}(0)) - 1)$$

If it is smaller than asympototic cut-off points -14.1 at 5% level, or -20.6 at 1% level then we suspect lack of convergence.

12.6 Eigenvalues

Eigen analysis of the kernel provides a mathematically rigourous approach towards converegence assesment. However, in general, this method is extremely complicated, and can be applied to only simple MC's. Here is an brief outline of the approach. We shall work with MC's with finite state spaces only.

Let K denote the transition matrix. K is a stochastic matrix, ie, the row sums are all 1, and the entries are all nonnegative. Thus, the vector of all ones is a right eigenvector corresponding to eigen value 1. Also, we know that any stationary distribution is a *left* eigenvalue corresponding to 1.

Theorem

K has all eigenvalues (possibly complex) in the unit disc.

Proof:

Let λ be an eigenvalue with left eigenvector u. Then

$$uK = \lambda u,$$

i.e,

$$\lambda u_i = \sum_j (u_j K_{ij}) \text{ for all } i.$$

Taking absolute values,

$$\lambda |u_i| \leq \sum_j (|u_j| K_{ij}) \text{ for all i.}$$

Adding over i, we get

$$\lambda \sum_i (|u_i|) \leq \sum_i (\sum_j (|u_j| K_{ij})).$$

Interchanging summations in the rhs, and using the stochastic nature of K, we see that

$$\lambda \sum_i (|u_i|) \leq \sum_j (|u_j|).$$

Hence the result. ♡

12.7 Bounding convergence rate using eigenvalues

We have already seen (using the definition) that any stationary distribution is a left eigenvector of K corresponding to eigenvalue 1. Let K be n by n, and let its eigenvalues be

$$\lambda_1, ..., \lambda_n,$$

where $\lambda_1 = 1$and

$$|\lambda_n| \leq ... \leq |\lambda_2| \leq 1.$$

For an irreducible and aperiodic Markov chain (with finite state space) there can be only one eigenvalue equal to 1, the rest being strictly smaller than 1 in absolute value, i.e.,

$$\lambda_1 = 1 and |\lambda_i| < 1 \text{ for all } i = 2, ..., n$$

Let v_i be a left eigenvector corresponding to λ_i for all i. For many Markov chains

$$\{v_1, ..., v_n\}$$

is a basis of $\mathbf{R}^n$. [Symmetric K is one such case, where the basis can also be chosen orthogonal.)

Now suppose our initial distribution is a vector $p(0)$. Since v_i's constitute a basis we can write $p(0)$ as their linear combination:

$$p(0) = a_1 v_1 + ... + a_n v_n$$

Multiplying both sides by K from the right, we get

$$p(1) = a_1 \lambda_1 v_1 + ... + a_n \lambda_n v_n$$

Similarly, we get

$$p(k) = a_1 (\lambda_1^k) v_1 + ... + a_n (\lambda_n^k) v_n$$

Since $|\lambda_i| < 1$ for $i = 2, ..., n$, hence all but the first term go to 0 as k goes to infinity. So

$$p(k) \to a_1 v_1, \text{ as } k \to \infty.$$

Since $p(k)$ is a probability vector for all k, hence so is the limit. Thus $a_1 v_1$ is the unique stationary distribution.

Now let's take a look at the other terms that all went to zero:

$$a_i (\lambda_i^k) v_i,$$

for $i = 2, ..., n$. While all these go to zero as k goes to infinity, the rates are different depending on $|\lambda_i|$. The smaller it is, the faster it goes to zero. So the λ_2 term will be the slowest one to vanish, and hence will determine the rate of convergence. The difference between this and 1 is called the **spectral gap** of the Markov chain.

$$\text{spectral gap } = 1 - |\lambda_2|.$$

Larger the spectral gap, better the convergence. There are theorems that bound the convergence rate exactly using powers of the spectral gap. However, those are beyond the present scope.