1 Markov Chains

First we present some basic definitions. A *stochastic process* is a parameterized collection of random variables. We will only consider stochastic processes where the parameter takes values in a countable space (e.g. the positive real numbers). Often we call such processes *discrete time* processes since the parameter is time in many applications. For us, the parameter will be location on the genome relative to some arbitrary zero, but we will still use the time terminology. A stochastic process has the *Markov property* if the future is conditionally independent of the past given the present. A stochastic process that has the Markov property is called a *Markov process*. A *Markov Chain* is a Markov process for which the random variables take only countably many values. The sample space of the individual random variables of a stochastic process is referred to as the *state space*, so a Markov Chain has a countable state space. A Markov Chain can have a parameter space that is not countable, but we will only need to consider discrete time Markov chains here. Moreover, we will only consider Markov Chains that have *finite* state spaces.

To specify a finite state space Markov Chain, one needs to specify a distribution for where the chain starts, and a set of conditional probabilities that specify how the chain moves from one state to another. Let X_t represent the value of the Markov Chain at time t. The set of conditional probabilities, $P(X_t = i_t | X_{t-1} = i_{t-1})$, where i are values in the state space of the Markov Chain and i_j denotes the state at time j, are conventionally stored in a matrix, called the *transition matrix*. The i, j element of the transition matrix gives the probability that the chain moves to state j at the next time given that the chain is in state i at the current time. The rows of such a matrix sum to one. If λ is a vector that holds the probability that the chain starts in each of the states, i.e. the i^{th} element of λ is the probability that the chain starts in state i, and if P is the transition matrix, then λP gives a vector whose i^{th} element is the probability that the chain is in state i at time 1. Similar reasoning leads one to conclude that λP^n gives a vector whose i^{th} element is the probability that the chain is in state i at time n (P^n is P matrix multiplied with itself n times).

A natural question to ask about Markov chains is what happens if they run for very long periods of time. First suppose that the Markov Chain is such that there is not a way to partition the state space into disjoint sets that can not be reached from one another. Such a Markov Chain is said to be *irreducible*. Next suppose that there is not a fixed sequence of states that the chain must pass through. Such a process is said to be *aperiodic*. If there exists a vector, π, such that $\pi P = \pi$, then we call π an *invariant measure*. If the elements of π sum to one then we call π an *invariant distribution*. For finite state space Markov chains, there always exists an invariant distribution: it is the left eigenvector associated with the eigenvalue 1 of the transition matrix. (A basic result from linear algebra is that transition matrices have an eigenvalue of 1 and all other eigenvalues are less than one.) The important theorem on convergence can now be stated: if the chain is irreducible, aperiodic and has an invariant distribution π, then $P(X_n = j) \to \pi_j$ for any initial distribution λ. For Markov chains with countable state space one must show that there exists an invariant measure in order to demonstrate convergence. There is also a sort of law of large numbers for Markov Chains, but such theorems are referred to as ergodic theorems in the Markov process context. For a very nice treatment of these topics (with accessible proofs) consult Norris (1997).

1.1 Statistical inference for discrete time finite state space Markov Chain

We can approach inference for Markov Chains with the techniques of maximum likelihood. Let θ denote a vector that holds the parameters involved in the model, i.e. the parameters in the initial distribution and the transition matrix. We first write the likelihood as the product of the conditional likelihoods, as we have

seen before, using the Markov property

$$L(x_1, \ldots, x_n|\theta) = L(x_1|\theta) \prod_{t=2}^{T} L(x_t|x_{t-1}, \theta).$$

Often the factor $L(x_1|\theta)$ is just ignored since interest is usually on the parameters in the transition matrix. (Alternatively, we may specify a prior distribution for the beginning state.) The factors $L(x_t|x_{t-1}, \theta)$ will represent observations on a multinomial observation since x_t takes only finitely many values. The success probabilities for this multinomial observation will come from the row of P that corresponds to the value of x_{t-1}. As analogy with the Binomial case, the MLEs for these multinomial probabilities will just be the sample proportions. Thus the MLE of the transition matrix is found by determining the sample proportions for each of the possible moves, i.e. the MLE of the i, j element of P is just the proportion of times that the Chain moved from state i to state j. If there is prior information about the elements of P these can be incorporated through a prior that multiplies the likelihood in the usual fashion. Since we have an expression for the likelihood, the usual techniques of hypothesis testing and confidence interval construction are at our disposal (we just need to compute the observed information).

2 Hidden Markov models

The fundamental idea behind a hidden Markov model is that there is a Markov process (i.e. a stochastic process with the Markov property) we can not observe that determines the probability distribution for what we do observe. Thus a hidden Markov model is specified by the transition density of the Markov chain and the probability laws that govern what we observe given the state of the Markov Chain. Given such a model, we want to estimate any parameters that occur in the model. We would also like to determine what is the most likely sequence for the hidden process. Finally we may want the probability density for the hidden states at every time point. We now turn to each of these problems.

First we introduce some notation. Let y_t represent the observed value of the process at time t for $t = 1, \ldots, T$, θ_t the value of the hidden process at time t and let ϕ represent parameters necessary to determine the probability distributions for y_t given θ_t and θ_t given θ_{t-1}. Our model is then described by the sets of probability distributions $p(y_t|\theta_t, \phi)$ and $p(\theta_t|\theta_{t-1}, \phi)$. The various distributions in which we are interested are $p(\phi|y_1, \ldots, y_T)$ (for parameter estimation), $p(\theta_t|y_1, \ldots, y_T)$ for all t (for inference about the hidden process) and $\text{argmax}_{\theta_1, \ldots, \theta_T} p(\theta_1, \ldots, \theta_T|y_1, \ldots, y_T)$ (the most likely path of the process). We will adopt a Bayesian perspective, so that we treat θ_t as a random variable. For the purposes of the next 2 sections, all probabilities are implicitly conditional on ϕ, i.e. we treat these as known. Later we will discuss estimation of ϕ.

2.1 Filtering and smoothing for hidden Markov models

We first treat the problem of finding $p(\theta_t|y_1, \ldots, y_t)$ for all t. This will be an important component to finding the desired probability distributions, and in the state space models literature is known as *filtering*. Now

$$\begin{aligned} p(\theta_t|y_1, \ldots, y_t) &\propto p(y_t|\theta_t, y_1, \ldots, y_{t-1}) p(\theta_t|y_1, \ldots, y_{t-1}) \\ &\propto p(y_t|\theta_t) \int p(\theta_t, \theta_{t-1}|y_1, \ldots, y_{t-1})\, d\theta_{t-1} \\ &\propto p(y_t|\theta_t) \int p(\theta_t|\theta_{t-1}) p(\theta_{t-1}|y_1, \ldots, y_{t-1})\, d\theta_{t-1}. \end{aligned}$$

So we have a recursion for the quantity $p(\theta_t|y_1,\ldots,y_t)$, hence we can update this quantity as time proceeds by using the 2 distributions that specify the model, $p(y_t|\theta_t)$ and $p(\theta_t|\theta_{t-1})$, and performing the necessary integration. For a finite state space Markov Chain, the integral is simply a sum over all possible values that θ_{t-1} can take. If the state space is not finite then the integral can be impossible to obtain analytically. An important exception is if both conditional distributions specified by the model are normal distributions. In this case one can evaluate the integrals: they turn out to by normal densities, and the recursion above simply becomes a recursion for the mean and variance of θ_t given all of the data up to that time. This recursion is called the Kalman filter and plays an important role in many applications. This recursion is often referred to as the *forward recursion* in the sequence alignment literature.

Now we will suppose we have run through all of the data using our forward recursion, and we will use these results to obtain the posterior distribution of θ_t given all of the observations up to time y_T. To this end

$$\begin{aligned}
p(\theta_t|y_1,\ldots,y_T) &\propto \int p(\theta_t,\theta_{t+1}|y_1,\ldots,y_T)\,d\theta_{t+1} \\
&\propto \int p(\theta_t|\theta_{t+1},y_1,\ldots,y_T)p(\theta_{t+1}|y_1,\ldots,y_T)\,d\theta_{t+1} \\
&\propto \int p(\theta_t|\theta_{t+1},y_1,\ldots,y_t)p(\theta_{t+1}|y_1,\ldots,y_T)\,d\theta_{t+1} \\
&\propto \int p(\theta_{t+1}|\theta_t,y_1,\ldots,y_t)p(\theta_t|y_1,\ldots,y_t)p(\theta_{t+1}|y_1,\ldots,y_T)\,d\theta_{t+1} \\
&\propto \int p(\theta_{t+1}|\theta_t)p(\theta_t|y_1,\ldots,y_t)p(\theta_{t+1}|y_1,\ldots,y_T)\,d\theta_{t+1}.
\end{aligned}$$

Hence we have a recursion for $p(\theta_t|y_1,\ldots,y_T)$ that runs in reverse time and involves $p(\theta_t|y_1,\ldots,y_t)$ which we calculated using the forward recursion (in addition to a distribution specified by the model). Once again, these integrals are interpreted as sums for finite state space Markov Chains. This recursion is referred to as the *backward recursion* in the sequence alignment literature and is called *smoothing* in other applications of state space models. So we obtain the posterior distribution of the state process at each time by running the forward recursion, saving these results, then running the backward recursion.

2.2 The posterior mode of the state sequence

Another property of the posterior distribution of interest is the posterior mode over the entire sample space of the state sequence. That is, we are interested in maximizing the function $p(\theta_1,\ldots,\theta_T|y_1,\ldots,y_T)$ with respect to the vector $(\theta_1,\ldots,\theta_T)$. This is not the same as finding the sequence of θ_t that maximize the univariate functions $p(\theta_t|y_1,\ldots,y_T)$. We will use dynamic programming to find this sequence. To this end we need a recursion for the maximum of the desired function.

Now

$$\begin{aligned}
\text{argmax}_{\{\theta_1,\ldots,\theta_t\}}p(\theta_1,\ldots,\theta_t|y_1,\ldots,y_t) &\propto \text{argmax}_{\{\theta_1,\ldots,\theta_t\}}p(y_t|\theta_1,\ldots,\theta_t,y_1,\ldots,y_{t-1})p(\theta_1,\ldots,\theta_t|y_1,\ldots,y_{t-1}) \\
&\propto \text{argmax}_{\{\theta_1,\ldots,\theta_t\}}p(y_t|\theta_t)p(\theta_t|\theta_{t-1})p(\theta_1,\ldots,\theta_{t-1}|y_1,\ldots,y_{t-1}).
\end{aligned}$$

This provides a recursion for the maximum and the sequence of θ_t for $t = 1,\ldots,T$ that achieves this maximum. Hence we use dynamic programming to sequentially update the maxima using this recursion. This algorithm is called the *Viterbi algorithm*.

2.3 Parameter estimation

In the previous 2 sections, we treated the parameters appearing in the conditional distributions that specify our hidden Markov model as known. In practice these parameters must be estimated from the data. We will know show how to compute the likelihood of these parameters. If there is prior information about the parameters, we could incorporate such information in the usual fashion.

Now, by the definition of conditional probability

$$p(\phi|y_1,\ldots,y_T) = \frac{p(\theta_1,\ldots,\theta_T,\phi|y_1,\ldots,y_T)}{p(\theta_1,\ldots,\theta_T|y_1,\ldots,y_T,\phi)}.$$

Hence we will have an expression for the posterior distribution of the model parameters if we can compute each of the 2 factors on the right hand side. For the numerator, we use Bayes theorem and the conditional independence of the y_ts given the θ_ts

$$\begin{aligned} p(\theta_1,\ldots,\theta_T,\phi|y_1,\ldots,y_T) &\propto p(y_1,\ldots,y_T|\theta_1,\ldots,\theta_T,\phi)p(\theta_1,\ldots,\theta_T,\phi) \\ &\propto p(\theta_1,\ldots,\theta_T,\phi)\prod_{t=1}^{T} p(y_t|\theta_t,\phi) \\ &\propto p(\phi)p(\theta_1|\phi)p(y_1|\theta_1,\phi)\prod_{t=2}^{T}\Big[p(y_t|\theta_t,\phi)p(\theta_t|\theta_{t-1},\phi)\Big]. \end{aligned}$$

For the denominator, we use the Markov property and manipulations identical to those used in the derivation of the backward recursion

$$\begin{aligned} p(\theta_1,\ldots,\theta_T|y_1,\ldots,y_T,\phi) &= p(\theta_T|y_1,\ldots,y_T,\phi)\prod_t p(\theta_t|\theta_{t+1},y_1,\ldots,y_T,\phi) \\ &= p(\theta_T|y_1,\ldots,y_T,\phi)\prod_t p(\theta_t|\theta_{t+1},y_1,\ldots,y_t,\phi) \\ &\propto p(\theta_T|y_1,\ldots,y_T,\phi)\prod_t p(\theta_{t+1}|\theta_t,\phi)p(\theta_t|y_1,\ldots,y_t,\phi). \end{aligned}$$

Therefore we can express the numerator and denominator in terms of the filtering densities, the transition probabilities between states and the probability distribution of the observations given the states. Thus, to evaluate the likelihood for ϕ at a given value of ϕ, we run the forward algorithm to get $p(\theta_t|y_1,\ldots,y_t,\phi)$ for all t, then use these along with the 2 conditional probabilities that specify the model in order to evaluate the likelihood. Note that the right hand side of the equation for the likelihood of ϕ depends on θ_t whereas the left hand side does not. This implies that the likelihood is given by this expression for any value of $\theta_1,\ldots,\theta_T$, hence for purposes of numerical stability, we evaluate the right hand side at the marginal modes $\operatorname{argmax} p(\theta_t|y_1,\ldots,y_t)$ since these are easy to evaluate after we have finished the forward recursion. We can then maximize this expression as a function of ϕ using any number of numerical maximization routines. This will give the MLE of ϕ.

2.4 Bayesian estimation of the state integrating out the model parameters

In practice, the distribution of the state sequence is of primary importance. Previously we treated ϕ as known, but now we have an expression for the posterior distribution of this parameter. If the sequence

is long (i.e. T is large) and there are not many parameters (i.e. ϕ is of low dimension) then treating the posterior mode (or MLE) of ϕ as the known value of ϕ is often a useful approximation. On the other hand, if there are many parameters and not much data, then ignoring uncertainty in the estimation of ϕ can lead to dramatic underestimation of the uncertainty in the estimation of the state process $\{\theta_t\}_{t=1}^T$. In this case we need to consider the uncertainty in the estimation of ϕ.

Although a number of approaches are possible, the following is typically easy to implement and works well compared to the popular alternatives. We will use a Bayesian simulation based approach. Our goal will be to obtain samples from the posterior distribution of the state process. We can then use these samples to characterize any aspect of the posterior distribution in which we are interested. For example, if we are interested in the posterior mean of θ_t for some t we just find the mean of our samples for θ_t. With enough samples, we can make this approximation to the mean as accurate as we desire.

Now

$$\begin{aligned} p(\theta_1, \ldots, \theta_T | y_1, \ldots, y_T) &= \int p(\theta_1, \ldots, \theta_T, \phi | y_1, \ldots, y_T)\, d\phi \\ &= \int p(\theta_1, \ldots, \theta_T | \phi, y_1, \ldots, y_T) p(\phi | y_1, \ldots, y_T)\, d\phi \end{aligned}$$

We can use this last equation as a basis for our simulations since we can evaluate the integral using Monte Carlo integration. That is, since we have an expression for the posterior distribution of ϕ we can draw samples from this distribution. Given a sample from the posterior of ϕ, we then simulate from $p(\theta_1, \ldots, \theta_T | \phi, y_1, \ldots, y_T)$ by using the recursion

$$p(\theta_1, \ldots, \theta_T | \phi, y_1, \ldots, y_T) = p(\theta_T | \phi, y_1, \ldots, y_T) \prod_t p(\theta_{t-1} | \theta_t, \phi, y_1, \ldots, y_T)$$

So we draw samples from each of the conditional distributions in turn conditional on the draw from the following value in the sequence. To draw a sample from the distribution $p(\theta_{t-1} | \theta_t, \phi, y_1, \ldots, y_T)$, we use the previously exploited relationship

$$p(\theta_{t-1} | \theta_t, \phi, y_1, \ldots, y_T) \propto p(\theta_t | \theta_{t-1}, \phi) p(\theta_t | y_1, \ldots, y_t, \phi).$$

(We showed this in the context of developing the backward recursion.) To use this expression to draw samples from the posterior, we draw the final state from its posterior, then draw samples from the other states conditional on the draw of the previous state. Since our state process has only finitely many states, we are just drawing from a multinomial distribution at each "time".

Thus the steps are

1. simulate many draws from the posterior distribution of ϕ
2. for each simulated ϕ run the forward recursion to obtain the necessary conditionals $p(\theta_t | y_1, \ldots, y_t, \phi)$
3. simulate the state process in reverse time as described above.

One can then calculate any aspect of the posterior distribution that is necessary for the scientific objective. For example, suppose one was interested in the first time that a state was reached with 95% certainty. This is easy to evaluate if we have many samples from the joint posterior distribution of all the states.

2.4.1 Simulating from the posterior of ϕ

While we have shown how to evaluate the marginal posterior distribution of the model parameters, ϕ, we did not describe how to draw simulations from this distribution. Since one can evaluate the distribution, one can draw a graph of the marginal distributions of each element of the vector ϕ. If these are approximately normal, as we expect on theoretical grounds, then we can approximate the posterior with a multivariate normal distribution (getting the covariance matrix from the negative inverse of matrix of second derivatives of the log posterior distribution). If there are only 2 or 3 elements in the vector ϕ, then one can calculate the posterior over a grid of values in the parameter space and sample from these conditionally, i.e. use $p(\phi) = p(\phi_1|\phi_2)p(\phi_2)$ where subscripts index elements of the vector ϕ.

Often the posterior of ϕ is not approximately normal and high dimensional, hence other methods are necessary to obtain simulations from this distribution. One very general way to draw samples from a probability distribution is to use the *Metropolis algorithm*. This algorithm proceeds by constructing a Markov Chain whose limiting distribution is the distribution from which you want samples. To use the algorithm, one simulates a long draw from the Markov chain, then uses samples from the chain once it has converged to its limiting distribution. In practice, determining whether the chain has converged to its limiting distribution is difficult. One sensible approach is to run several independent chains and examine how long it takes for them to mix (an approach I often use).

But how can one construct a Markov chain whose limiting distribution is $p(\phi|y_1, \ldots, y_T)$? This turns out to be surprisingly simple. Suppose we have some distribution over the sample space from which we can draw simulations given the current value of the chain. Call this distribution the jumping distribution, and denote it $J(\phi|\phi_t)$. We require that jumping distributions are symmetric in the sense that $J(\phi_a|\phi_b) = J(\phi_b|\phi_a)$, but an extension of the Metropolis algorithm called the Metropolis-Hastings algorithm allows one to use non-symmetric jumping distributions. For example we could use a normal density with mean given by ϕ_t and variance given by the negative inverse of the matrix of second derivatives of the log posterior distribution (an approach that often works quite well). Normal densities clearly have the desired symmetry. We need to choose some initial value for our chain, denote it ϕ_0. The algorithm proceeds by drawing a simulation from the jumping distribution, call it ϕ^*, then evaluating the ratio, r, of the posterior at ϕ^* to the posterior at the current location of the chain, ϕ_{t-1}

$$r = \frac{p(\phi^*|y_1, \ldots, y_T)}{p(\phi_{t-1}|y_1, \ldots, y_T)}.$$

Then $\phi_t = \phi^*$ with probability $\min(r, 1)$ otherwise $\phi_t = \phi_{t-1}$.

To see that this algorithm will indeed draw samples from the limiting distribution of the chain, first suppose that the state space and jumping distributions are such that the chain is aperiodic and irreducible. These 2 conditions are typically met in most applications. Then there exists a limiting distribution for the chain. We now show that this limiting distribution is the posterior distribution we seek. To this end, consider the joint distribution of 2 adjacent values in the chain, $p(\phi_t, \phi_{t-1})$ (implicitly conditioning on the data). Using $p(y)$ to denote the posterior density at y, suppose $p(x) > p(y)$. Then

$$p(\phi_t = x, \phi_{t-1} = y) = p(y)J(x|y),$$

and also

$$p(\phi_t = y, \phi_{t-1} = x) = p(x)J(y|x)r,$$

where $r = \frac{p(y)}{p(x)}$. But then

$$p(\phi_t = y, \phi_{t-1} = x) \quad = \quad p(y)J(y|x)$$

$$= \quad p(y)J(x|y),$$

by symmetry of the jumping distribution. But then

$$\begin{aligned} p(\phi_t = x) &= \sum_y p(\phi_t = x, \phi_{t-1} = y) \\ &= \sum_y p(\phi_t = y, \phi_{t-1} = x) \\ &= p(\phi_{t-1} = x). \end{aligned}$$

This implies that the marginal distributions at each time point are the same. Therefore if ϕ_{t-1} is a draw from the posterior distribution, then so is ϕ_t. Note that these will not be independent draws (it is a Markov Chain).

2.5 Using the Gibbs sampler to obtain simulations from the joint posterior

Perhaps the most common method for obtaining samples from the posterior distribution of the states and model parameters is to use the *Gibbs sampler*. The Gibbs sampler is a special case of the Metropolis-Hastings algorithm. The idea of the Gibbs sampler is to simulate each parameter conditional on the values of all other parameters. For example, for hidden Markov models, one would sample the state process given the model parameters as we have outlined above, then simulate the model parameters conditional on the values of the state. It is usually simple to simulate from the posterior distribution of the model parameters given the values of the state process because this is identical to parameter estimation for fully observed Markov models. Consult Gelman et al. (1995) for the proof that the Gibbs sampler is a special case of the Metropolis-Hastings algorithm.

The drawback of using the Gibbs sampler comes from the fact that we are using the Metropolis algorithm over a much larger sample space compared to the approach advocated above. In practice, it is difficult to demonstrate that a chain has converged to its limiting distribution, and this problem becomes more difficult as the number of parameters increases and as the correlation between parameters in the posterior distribution increases. For hidden Markov models, typically T is large (e.g. 1000) and the number of model parameters is small (e.g. 20). Hence the Gibbs sampler constructs a Markov chain over a very large parameter space (e.g. 1020) compared to the method that first simulates the model parameters then simulates the state without a Markov chain (e.g. 20). In addition, the correlation between adjacent states in the posterior distribution is typically quite high (since they represent values of a Markov chain).

References

Norris, J. (1997), *Markov Chains*, Cambridge University Press.

Exercises

1. Consider a 2 state Markov chain with transition probabilities p_1 and p_2. Find the limiting distribution of the chain.

2. Suppose we observe a Binomial random variable, y_t, with fixed sample size n over time that has one of 2 success probabilities, ϕ_1 and ϕ_2, and the probability of success is determined by a 2 state hidden Markov chain with transition probabilities ϕ_3 and ϕ_4. Give explicit forms for the forward and backward recursions.

3. Use the proof that the limiting distribution of the Metropolis algorithm is the posterior distribution to determine the appropriate choice of r for the Metropolis-Hastings algorithm. This is the only difference between the Metropolis algorithm and the Metropolis-Hastings algorithm.