1 An introduction to Stochastic Processes

1.1 The Poisson process

We model a situation where events occur spontaneously, at random, in time. Examples are the emissions of α-particles in a radioactive experiment, arrivals of customers at a post office, the passing of cars on a quiet road. The important feature of these events is that they are unpredictable (they occur 'at random').

The average rate λ at which events occur is constant over time (not true for post office customers over a whole day, but good enough for a half hour period).
The occurence of events after time t is independent of what happened up to time t.
We also assume that events can only occur singly (never ≥ 2 simultaneously).

$X(t)$, the number of occurrences of a Poisson process at time t, has a Poisson distribution with parameter λt.

e.g. Fax messages arrive at an office at the mean rate of three per hour according to a Poisson process.
(i) What is the probability that exactly two messages are received between 9.00 and 9.40 ?
(ii) What is the probability that no messages arrive between 10.00 and 10.30 ?
(iii) What is the probability that not more than three messages are received between 10.00 and 12.00 ?

Let an hour be the unit of time that we work with. Thus we have a Poisson process of rate $\lambda = 3$. The number of messages in time t is thus Poisson $(3t)$.
(i) 9.00 to 9.40 - the number of calls is Poisson $(3 \times 2/3 = 2)$, so

$$P(2messages) = e^{-2}\frac{2^2}{2!} = 0.271$$

(ii) 10.00 to 10.30 - the number of calls is Poisson $(3 \times 0.5 = 1.5)$, so
P(0 messages) $= e^{-1.5} = 0.223$

(iii) 10.00 to 12.00 - the number of calls is Poisson $(3 \times 2 = 6)$.
$P(X \leq 3) = P(X = 0) + P(X = 1) + P(X = 2) + P(X = 3) =$

$$e^{-6}\frac{6^0}{0!} + e^{-6}\frac{6^1}{1!} + e^{-6}\frac{6^2}{2!} + e^{-6}\frac{6^3}{3!} = 0.151$$

If T is the time until the next event from a given starting point (since the process is 'memoryless' it does not matter when this starting point is), then T has an exponential distribution with parameter λ.

e.g. In the previous example
(iv) What is the probability that the first message after 10.00 occurs before 11.00 ?

This is a Poisson process with rate 3, $t = 1$, and so $P(T \leq 1) = 1 - e^{-3\times1} = 0.9502$

1.2 Birth processes

1.3 The simple birth process

Consider a population where each individual alive in the population generates further offspring according to a Poisson process at rate β (new individuals are produced asexually). We assume that the initial population size is x_0 and that there are no deaths, so that the population increases with time. It can be shown that the distribution of $X(t)$, the number of individuals alive at time t, follows a negative binomial distribution with parameters $e^{-\beta t}$ and x_0.

Note that if $x_0 = 1$, this becomes the geometric distribution with parameter $e^{-\beta t}$.

e.g. A population starts at time 0 with a single individual. Let the birth rate be two per week.
(i) What is the probability that after three weeks there are exactly two individuals?
(ii) What is the probability that after one week there are between two and four individuals (inclusive)?

(i) $x_0 = 1$, $\beta = 2$ per week, and $t = 3$ i.e.

$$p_2(3) = \tbinom{1}{0}e^{-2\times3}(1 - e^{2\times3})^1 = e^{-6}(1 - e^{-6}) = 0.00247$$

(ii) $x_0 = 1, \beta = 2, t = 1. P[2 \leq X \leq 4] = P(X = 2) + P(X = 3) + P(X = 4) =$
$e^{-2}(1 - e^{-2}) + e^{-2}(1 - e^{-2})^2 + e^{-2}(1 - e^{-2})^3 = 0.3057$

1.4 The pure death process

We consider a population in which there are no births, just deaths. Observations start with x_0 individuals alive at time 0 - these individuals die independently of each other, and eventually the population dies out completely.

This model is approached best by considering every individual separately. The probability that an individual is alive at time t, $P_a(t) = Ae^{-\nu t} = e^{-\nu t}$. We can use the binomial theorem to deduce that the probability that j individuals are still alive at time t, is given by

$$p_j(t) = (e^{-\nu t})^j (1 - e^{-\nu t})^{x_0 - j} \binom{x_0}{j}$$

In particular the probability that the population is extinct by time t is

$$p_0(t) = (1 - e^{-\nu t})^{x_0}$$

e.g. A population starts at time 0 with 4 individuals. The population follows a pure death process at a rate of 1 every 2 days.
(i) Find the probability that there is exactly one individual alive after a week.
(ii) Find the probability that the population has died out after a week.
(iii) Find the probability that the population has died out after two weeks, given that the total number of survivors after one week was 2.

(i) $t = 7, \nu = 0.5$ i.e.

$$p_1(7) = e^{-3.5}(1 - e^{-3.5})^3 \binom{4}{1} = 0.1102$$

(ii) $t = 7, \nu = 0.5 \Rightarrow$

$$p_0(7) = (1 - e^{-3.5})^4 = 0.8846$$

(iii) The process is memoryless, so that
P(0 after 2 weeks/ 2 after 1 week) = P(0 after 1 week/ 2 after 0 weeks) =

$$(1 - e^{-3.5})^2 = 0.9405$$

1.5 Birth and death processes

We shall now consider a population model, similar to those of the last chapter, but with both births and deaths. In the Poisson process and the simple birth process the only change possible is an increase in the population. In the pure death process the population could only be reduced. Here both may occur.

1.6 The simple birth-death process

In the previous section we considered the simple birth process and the pure death process. Now we shall combine the two.

In the simple birth process, each individual gives birth at rate β, so that when the population is of size x, the birth rate is βx. In the pure death process individuals die at rate ν, so that the death rate is νx. We could look for an expression for $X(t)$, the number of individuals alive at time t. This turns out to be quite complicated to do; we can answer some simpler questions more easily.

For instance, what is the probability that the population eventually becomes extinct? If $\beta \leq \nu$, the population is certain to become extinct. If $\beta > \nu$, the probability of eventual extinction is

$$\left(\frac{\nu}{\beta}\right)^{x_0}$$

If we are not interested in the time that a particular event occurs, but only in its type (is it a birth or a death), we can simplify the model. Relabelling the time of the occurrence of the ith event as i, we obtain a new random process $\{X_i; i = 0, 1, 2 \ldots\}$ in discrete time. This process is said to be *embedded* in the original process, i.e. is an *embedded process.* X_i is the size of the embedded process at time i, and of the population immediately after the ith change.

$$p = \frac{\beta}{\beta + \nu}$$

where p is the probability that any particular event is a birth. We have what is called a simple random walk with

$$P[X_i = x + 1/X_{i-1} = x] = \frac{\beta}{\beta + \nu}$$

and an absorbing barrier at zero. This means that we can use standard results to show;

the expected number of events to extinction is

$$\frac{x_0(\nu + \beta)}{\nu - \beta} \qquad \beta < \nu$$

and is infinite if $\beta \geq \nu$.

The probability that the size of our population reaches m individuals at some point (before possibly becoming extinct), has probability

$$\frac{1 - \left(\frac{\nu}{\beta}\right)^{x_0}}{1 - \left(\frac{\nu}{\beta}\right)^{m}} \qquad \beta \neq \nu$$

$$\frac{x_0}{m} \qquad \nu = \beta$$

e.g. If a simple birth-death process starts with 5 individuals, what is the probability that it reaches 10 given that it becomes extinct if
a) $\beta = 4, \nu = 6$?
b) $\beta = 9, \nu = 6$?

a) P(reaches 10) =

$$\frac{1 - \left(\frac{6}{4}\right)^5}{1 - \left(\frac{6}{4}\right)^{10}} = 0.1164$$

The process is certain to become extinct, so that P(reaches 10 and becomes extinct) = P(reaches 10) = 0.1164

b) P(reaches 10) =

$$\frac{1 - \left(\frac{6}{9}\right)^5}{1 - \left(\frac{6}{9}\right)^{10}} = 0.8836$$

P(reaches 10 / becomes extinct)=

P(extinction/reaches 10)P(reaches 10)/P(extinction)

P(extinction)=

$$\left(\frac{6}{9}\right)^5 = 0.1317$$

P(extinction/reaches 10)=

$$\left(\frac{6}{9}\right)^{10} = 0.0173$$

Thus P(reaches 10 /becomes extinct)=

$$\frac{0.0173 \times 0.8836}{0.1317} = 0.116$$

So if we know that the population becomes extinct, it is unlikely to have reached 10 before doing so.

2 Markov Chains

in this section we shall discuss Markov chains with a finite number of states. this will be central to a lot of what follows in later sections. A Markov chain is defined by a number of states
$E_1, \dots, E_n$
, one of which is occupied at a given time. We follow the process from time
$t = 1, 2, 3, \dots$
As time moves a step forward, the process moves from one state to another (possibly staying at the same state) following some straightforward probabilistic rules.

The process is memoryless and homogeneous in time. Thus the probability of moving from E_i at time t to E_j at time $t+1$ takes the same value for all t. So only the current state matters in determining future movements; the earlier history of the process is irrelevant.

2.1 The transition matrix

If we are in state E_i then the probability that the process moves to state j in the next time step is labelled p_{ij} for each pair of states i and j. For convenience we group all of these transition probabilities together into a single matrix, labelled P.

$$\begin{array}{|c|ccccc|} & E_1 & E_2 & E_3 & \dots & E_n \\ \hline E_1 & p_{11} & p_{12} & p_{13} & \dots & p_{1n} \\ E_2 & p_{21} & p_{22} & p_{23} & \dots & p_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ E_n & p_{n1} & p_{n2} & p_{n3} & \dots & p_{nn} \end{array} \tag{1}$$

the entry in row i and column j os p_{ij}, the probability of moving form i to j. Since wherever the process stands at time t, it must end up at precisely one state at time $t+1$, the probabilities in every row must sum to 1, i.e.

$$\sum_{j=1}^{n} p_{ij} = 1$$

for every value of i.

It is easy to see that the random walk from the previous section (where at each step except 0 there was a probability p of increasing by 1 and $q = 1 - p$ of decreasing by 1) has transition matrix

$$
\begin{array}{c|cccccccc|}
 & 0 & 1 & 2 & \dots & k-1 & k & k+1 & \dots \\
\hline
0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots \\
1 & q & 0 & p & \dots & 0 & 0 & 0 & \dots \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
k & 0 & 0 & 0 & \dots & q & 0 & p & \dots \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots
\end{array}
\tag{2}
$$

2.2 Absorbing states

The random walk example that we discussed above is a Markov chain with an *absorbing state*, since the state '0' cannot be left when it is reached. Such a Markov chain is easily recognised; if its transition matrix has a '1' on its leading diagonal, then the corresponding state is an absorbing one; if there are no 1s on the leading diagonal, then it does not contain an absorbing state.

For a Markov chain with at least one absorbing state, sooner or later one of its absorbing states will be entered and never left. We may wish to consider what is the probability that we enter a given state, or the expected time to entering a state.

For example, the probability that the size of the population from our birth and death process from Section 11 reaches m individuals at some point (before possibly becoming extinct), was given to be

$$\frac{1-\left(\frac{\nu}{\beta}\right)^{x_0}}{1-\left(\frac{\nu}{\beta}\right)^{m}} \qquad \beta \neq \nu$$

This was found by using the random walk transition matrix from above, with the single change that m becomes an absorbing state, so that row m is all 0s except $p_{mm} = 1$.

If a Markov chain has no absorbing states, the questions that we can ask about it are more varied. To simplify matters, and because these are the key properties of the processes used in Monte Carlo Markov Chains from later sections, we concentrate on Markov chains which are *finite*, *aperiodic* and *irreducible.* A chain is finite if it has a finite number of possible states (as opposed to the birth and death process). A chain is aperiodic if there is no state that can be returned to at regular intervals only (for instance even time points only). For instance

$$\begin{vmatrix} 0 & 0 & 0.6 & 0.4 \\ 0 & 0 & 0.3 & 0.7 \\ 0.5 & 0.5 & 0 & 0 \\ 0.2 & 0.8 & 0 & 0 \end{vmatrix} \tag{3}$$

is periodic. If a Markov chain has no 0s down its leading diagonal, then it is aperiodic. A chain is irreducible if whatever state we are in, we can reach some other state from that state from some path. For instance

$$\begin{vmatrix} 0.2 & 0.2 & 0.2 & 0.4 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0.6 & 0.4 \end{vmatrix} \tag{4}$$

is reducible, since once states 3 or 4 have been reached, the process can never return to states 1 or 2.

2.3 Stationary distributions

Suppose that a Markov chain has a transition matrix P which has probability ϕ_j of being in state E_j at time t, $j = 1, \dots, n$. Suppose that

$$\phi_j = \sum_{i=1}^{n} \phi_i p_{ij} \qquad j = 1, 2, \dots, n$$

so that the probability of being in state j at time $t+1$ is still ϕ_j for all j. The probability distribution $(\phi_1, \phi_2, \dots, \phi_n)$ is *stationary.*

To find a stationary distribution, we must solve the set of equations above. For example, for the matrix

$$\begin{vmatrix} 0 & 0 & 0.6 & 0.4 \\ 0 & 0 & 0.3 & 0.7 \\ 0.5 & 0.5 & 0 & 0 \\ 0.2 & 0.8 & 0 & 0 \end{vmatrix} \tag{5}$$

we obtain
$\phi_1 = 0.5\phi_3 + 0.2\phi_4 \ldots (i)$
$\phi_2 = 0.5\phi_3 + 0.8\phi_4 \ldots (ii)$
$\phi_3 = 0.6\phi_1 + 0.3\phi_2 \ldots (iii)$
$\phi_4 = 0.4\phi_1 + 0.7\phi_2 \ldots (iv)$

The first two equations yield
$0.6\phi_4 = \phi_2 - \phi_1 \ldots (v)((ii) - (i))$
$1.5\phi_3 = 4\phi_1 - \phi_2 \ldots (vi)(4(i) - (ii)$
$1.24\phi_1 - 0.58\phi_2 = 0 \Rightarrow \phi_2 = 2.1379\phi_1(0.6(iv) - (v)$
Thus from (v) $\phi_4 = 1.8965\phi_1$
and from (vi) $\phi_3 = 1.2414\phi_1$
So $\phi_1(1 + 2.1379 + 1.2414 + 1.8965) = 1 \Rightarrow \phi_1 = 0.1593$ giving
$\phi_2 = 0.3407, \phi_3 = 0.1978, \phi_4 = 0.3022$

Notice that we could have solved this more easily by observing that since the process alternates between states 1,2 and 3,4 we must have $\phi_1 + \phi_2 = \phi_3 + \phi_4 = 0.5$

The stationary distribution of

$$\begin{vmatrix} 0.2 & 0.2 & 0.2 & 0.4 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0.6 & 0.4 \end{vmatrix} \tag{6}$$

can be found easily since it is clear that $\phi_1 = \phi_2 = 0$. Sovling for the other two strategies, gives $\phi_3 = 6/11, \phi_4 = 5/11$

2.4 Markov chains in continuous time

A Markov chain in continuous time can be defined in terms of the transition rates between the states, so that the rate from state E_i to E_j is given by q_{ij}. In fact each of the models from Section 11 are continuous time Markov chains. if we are not worried about the specific times of events, but just the sequence, we can construct an embedded process in precisely the same way as we did in that section, which is then a Markov process in discrete time, with a transition matrix given by

$$p_{jk} = \frac{q_{jk}}{\sum_{i=1,i\neq j}^{n} q_{ji}}$$

Exercise: Show this.

Stationary distributions are found in the same way as before. In fact, the equations become

$$\phi_j = \sum_{i=1}^{n} \phi_i q_{ij} \qquad j = 1, 2, \dots, n$$

So for the birth and death process of Section 11, for $j > 0$, $q_{j,j+1} = \beta, q_{j,j-1} = \nu$ and $q_{ji} = 0$ otherwise, and we obtain

$$p_{j,j+1} = \frac{q_{j,j+1}}{\sum_{i=1}^{n} q_{ji}} = \frac{\beta}{\beta + \nu}$$

$$p_{j,j-1} = \frac{q_{j,j-1}}{\sum_{i=1}^{n} q_{ji}} = \frac{\nu}{\beta + \nu}$$

and $p_{ji} = 0$ otherwise

3 DNA sequence analysis

3.1 Analysing a single sequence

Before we can analyse a sequence, we need to determine what the sequence is. It turns out that it is not possible to accurately identify long DNA sequences. Rather, many overlapping small sequences are taken (of the order of 500 bases). Then these pieces need to be assembled into a single long sequence by matching overlapping regions on to each other; such an assemblage is called a *contig.* To do this a sufficiently large collection of such short subsequences need to be collected. This is called *shotgun sequencing.* A sequence has *nX coverage* if for the original sequence of length G, the total length of all the fragments is nG. To expect to cover 99% of the sequence, it can be shown that we need to have $4.6X$ coverage. This is the technique that has been used to sequence the whole human genome.

FIGURE 13.1

Suppose that the DNA sequence that we are mapping has length G, and we have N fragments each of length L, then the coverage is given by

$$a = \frac{NL}{G}$$

The mean proportion of the sequence covered by one or more fragments is the probability that a random point has its left hand end in the interval of length L immediately to the left of this point.

FIGURE 13.2

On the (reasonable) assumption that L is a lot smaller than G, the number of fragments whose left-hand edge falls in this region is Binomial with parameters N and L/G. The probability that no fragment lies in this interval is thus

$$\left(1 - \frac{L}{G}\right)^N \approx e^{-a}$$

The probability that the point is covered is thus $1 - e^{-a}$. When $a = 4.6$ this leads to a probability of 0.99, as above.

3.2 Modelling of DNA sequences

The structure of DNA can be thought of as long sequences of nulcletides of four types labelled a, g, c and t. We are interested in the structure of these sequences, based upon the relationship between the type of nucleotide which appears at neighbouring sites. The sequences comprise different regions, where the structure of the components can vary from region to region. The simplest 'structure' conceivable is complete independence, where the nucleotide at every site is independent of all others, and identically distributed with them. A more plausible model is based upon a Markov structure; namely that neghbouring nucleotides are correlated, but that only immediate neighbours influence the distribution at a site. The following table shows the kind of data that we are interested in for the Markov model.

	a	g	c	t	Total
a	Y_{11}	Y_{12}	Y_{13}	Y_{14}	$Y_{1.}$
g	Y_{21}	Y_{22}	Y_{23}	Y_{24}	$Y_{2.}$
c	Y_{31}	Y_{32}	Y_{33}	Y_{34}	$Y_{3.}$
t	Y_{41}	Y_{42}	Y_{43}	Y_{44}	$Y_{4.}$
Total	$Y_{.1}$	$Y_{.2}$	$Y_{.3}$	$Y_{.4}$	Y

Such data can be used to estimate the transition matrix which defines our Markov model. In particular we can test whether the transition probabilities are identical for each nucleotide using a χ^2 test as used for standard categorical data.

$$Q = \sum_{i=1}^{4}\sum_{j=1}^{4} \frac{(Y_{ij} - Y_{i.}Y_{.j}/Y)^2}{Y_{i.}Y_{.j}/Y}$$

will have an approximate χ^2 distribution with $(4-1)(4-1) = 9$ degrees of freedom under the null hypothesis of identical transition probabilities.

We may be interested in the possibility of long repeats of a particular nucleotide, say a. Suppose that for a particular DNA sequence, the probability of a appearing at any given site is p. After any given occurrence of a nucleotide other than a (a 'failure') there will be a run of Y a's ('successes'); this length can be zero if a second failure follows the first. Y has a geometric distribution with parameter p, i.e.

$$P(Y = y) = (1-p)p^y$$

It can be shown that if n such (independent) sequences occur, and Y_{max} is the largest of these, then

$$P(Y_{max} \geq y) = 1 - (1-p^y)^n$$

In a DNA sequence of length N, the number of such sequences of a's is approximately $n = N(1-p)$ (the number of failures) so that

$$P(Y_{max} \geq y) = 1 - (1-p^y)^{N(1-p)}$$

This gives the p-value of the test of the null hypothesis that every successive nucleotide is independent.

e.g. if $N = 100000, p = 0.25$ and the observed $y_{max} = 10$, then P is as follows:

$$P = 1 - (1 - 0.25^{10})^{75000} = 0.0690272$$

3.3 Comparing two DNA sequences

Suppose that we have two DNA sequences. One way of comparing the two is to test whether the proportion of each nucleotide is the same in each sequence. This follows a typical χ^2 test for tabular data as above, where the table is

	a	g	c	t	Total
S1	Y_{11}	Y_{12}	Y_{13}	Y_{14}	$Y_{1.}$
S2	Y_{21}	Y_{22}	Y_{23}	Y_{24}	$Y_{2.}$
Total	$Y_{.1}$	$Y_{.2}$	$Y_{.3}$	$Y_{.4}$	Y

In this case there are $(2 - 1) \times (4 - 1) = 3$ degrees of freedom.

We are often interested in finding *alignments* between two sequences. For example, suppose that we have the sequences *cgggtatccaa* and *ccctaggtccca* which may be descended from a single common ancestral sequence (changes occur, by means of substitutions, deletions and insertions). We try to find the best match up between the two sequences, where gaps can be introduced into either sequence (as deletions and insertions alter the position of whole sub-sequences).

The following is an example of an alignment between the two sequences
c g g g t a $-$ $-$ t c c a a
c c c $-$ t a g g t c c c a

How do we decide whether two sequences are well aligned. one way is to give an alignment score to any alignment between two sequences. For example we can use

number of matches-number of mismatches-number of gaps

It makes sense to penalise the use of gaps as this gives us a lot more options in our sequences, and we could get a perfect match if allowed to use an unlimited number of gaps for free. Scoring schemes can be more complicated, of course, but in some cases this one works well.

Once we have a scoring system, we can compare the possible alignments and choose the one with the highest score. If no gaps are allowed, then we can do this for every possible alignment for 2 sequences. The use of gaps makes this not feasible; we need to use an algorithmic method. One such algorithm is the Needleman-Wunsch algorithm. This is an example of a dynamic programming algorithm. We will come back to this and related algorithms at the end of the course.

For example, if we use the above scoring mechanism on the alignment
c t t a g − g − −
c a t − g a g a a
the score is $4 - 1 - 4 = -1$.

One related alignment problem is the *linear gap model.* Given two sequences, we try to find the subsequence of the longer one that can be aligned with the shorter one in the best way (where gaps are allowed).

Let $x = X_1X_2 \dots X_m$ be the shorter sequence and $y = Y_1Y_2 \dots Y_n$ be the longer one. Also denote the subsequence of y given by $Y_kY_{k+1} \dots Y_j$ as $y_{k,j}$. To get a value of $B(x, y_{k,j})$, the score of the best alignment that we can find between x and $y_{k,j}$, the running time of the Needleman-Wunsch algorithm is of the order of $m(j-k)$.

We want to find the best alignment overall, given by

$$max(B(x, y_{k,j}) : 1 \leq k \leq j \leq n)$$

If we did this exhaustively, it would take time of the order of mn^3. It is possible to use another algorithmic approach to improve this considerably, to the order of mn.

3.4 Protein sequences and substitution matrices

When looking at DNA sequences, such simple scoring systems are often effective. However for protein sequences, some substitutions are much more likely than others; an alignment algorithm that takes this into account is a lot more effective. There are two main type of substitution matrix, PAM (Accepted Point Mutation) and BLOSUM (BLOcks SUbstitution Matrices) with rather contrived acronyms, as you can see. We will consider PAM substitution matrices only.

An accepted point mutation is the substitution of one amino acid by another that is ‘accepted’ by evolution, so that effectively this change happens for the whole of a given species. A PAM1 transition matrix is the Markov transition matrix applying to the time period in which we expect 1% of amino acids to undergo accepted point mutations within the given species.

We can find transition matrices for larger distances by raising the matirx to a given power; transitions for n PAM units is given by PAMn,
$M_n = M_1^n$

Suppose that all amino acids are equally frequent (so that $p_j = 0.05$), that all are equally likely to be substituted by some other amino acid in any given time, and all substitutions are equally likely. Then the PAM1 matrix $M_1 = (m_{ij})$ is given by,

$$m_{ii} = 0.99, m_{ij} = \frac{0.01}{19} \quad i \neq j$$

For this matrix it can be shown that these probabilities in the PAMn matrix become

$$m_{ii}^{(n)} = 0.05 + 0.95(94/95)^n$$

$$m_{ij}^{(n)} = 0.05 - 0.05(94/95)^n$$

For $n = 10$ for example, we obtain the probability that any given amino acid is unchanged as

$$m_{ii}^{(10)} = 0.05 + 0.95(94/95)^{10} = 0.9046$$

and the probability that it is replaced by any given alternative amino acid is

$$m_{ij}^{(10)} = 0.05 - 0.05(94/95)^{10} = 0.00502$$

Of course for real populations these probabilities are not so evenly spread, as some amino acids are more common than others, and transition rates vary between sites etc. Data can be used to estimate the values in the PAM1 matrix.

Question: If we find amino acid, A, at a point in the sequence, what is the probability that 2 PAM units ago it was a B?

4 Monte Carlo Markov Chains I

In this section we shall introduce a variant on the Markov chain model and consider some computational statistical techniques which will lead us to the mainstream Monte Carlo Markov chain methods of the next section. The methods of this section, both theoretical and practical, have a wider applicability as well.

4.1 Hidden Markov models

A Hidden Markov model is an extension of the Markov chain idea that we met in the previous sections. Suppose that we have a Markov chain where states are visited according to a transition matrix P, so that a sequence of states are visited which we label $q_1, q_2, q_3 \ldots$. At each state a symbol is emitted from some collection of possibilities, so that we have a sequence of symbols $O_1, O_2, O_3, \ldots$. We can label the sequence of q_is by $\mathbf{Q}$ and the sequence of O_is by $\mathbf{O}$. Often we know the sequence $\mathbf{O}$ but we do not know the underlying sequence of states $\mathbf{Q}$; the sequence $\mathbf{Q}$ is *hidden.*

We may be able to estimate what the underlying sequence $\mathbf{Q}$ is from the information that we have in $\mathbf{O}$, if each state does not emit the symbols with the same probabilities as all of the other states. To do this we thus need two pieces of information; the transition matrix P and a collection of probability distributions

Example 14.1. Consider the Markov chain with transition matrix

$$\begin{vmatrix} 0.9 & 0.1 \\ 0.8 & 0.2 \end{vmatrix} \tag{7}$$

where the process is equally likely to start in either state. There are only two possible symbols that can be emitted 1 and 2; state S_1 emits 1 with probability 0.5 (and so 2 with probability 0.5), state S_2 emit 1 with probability 0.25 (and so 2 with probability 0.75).

Suppose that we observe the sequence 2,2,2. What is the most plausible underlying sequence? There are 8 possibilities, each of which is written below with the probability of obtaining the observed sequence AND underlying sequence calculated (i.e. $P(\mathbf{Q} \bigcap \mathbf{O})$).

$$S_1 \to S_1 \to S_1 \quad 0.5 \times 0.5 \times 0.9 \times 0.5 \times 0.9 \times 0.5 = 0.0506$$

$$S_1 \to S_1 \to S_2 \quad 0.5 \times 0.5 \times 0.9 \times 0.5 \times 0.1 \times 0.75 = 0.0084$$

$$S_1 \to S_2 \to S_1 \quad 0.5 \times 0.5 \times 0.1 \times 0.75 \times 0.8 \times 0.5 = 0.0075$$

$$S_1 \to S_2 \to S_2 \quad 0.5 \times 0.5 \times 0.1 \times 0.75 \times 0.2 \times 0.75 = 0.0028$$

$$\begin{array}{ll}
S_2 \to S_1 \to S_1 & 0.5 \times 0.75 \times 0.8 \times 0.5 \times 0.9 \times 0.5 = 0.0675 \\
S_2 \to S_1 \to S_2 & 0.5 \times 0.75 \times 0.8 \times 0.5 \times 0.1 \times 0.75 = 0.0113 \\
S_2 \to S_2 \to S_1 & 0.5 \times 0.75 \times 0.2 \times 0.75 \times 0.8 \times 0.5 = 0.0225 \\
S_2 \to S_2 \to S_2 & 0.5 \times 0.75 \times 0.2 \times 0.75 \times 0.2 \times 0.75 = 0.0084
\end{array}$$

The best sequence maximises

$$P(\mathbf{Q}|\mathbf{O}) = \frac{P(\mathbf{Q} \bigcap \mathbf{O})}{P(\mathbf{O})}$$

which is equivalent to maximising the above. Thus the most plausible sequence is S_2, S_1, S_1.

An HMM consists of the following five components

(1) A set of n states $S_1, \ldots, S_n$
(2) An alphabet of distinct observation symbols $A = \{a_1, \ldots, a_M\}$
(3) The transition probability matrix $P = (p_{ij})$ where

$$p_{ij} = P(q_{t+1} = S_j | q_t = S_i)$$

(4) Emission probabilities for each state: if the process is in state S_i then $b_i(a_j) = P(S_i \text{ emits } a_j)$, so $\sum_{j=1}^{M} b_i(a_j) = 1$
(5) An initial distribution of states $\pi_i = P(q_1 = S_i)$

Components 1,2 and 5 define the underlying Markov chain.

Thus in our example $n = 2$, $M = 2$ with $a_1 =$ '$1'$, $a_2 =$ '$2'$, P is the 2×2 matrix (e.g. $p_{12} = 0.1$), $b_1(a_1) = 0.5, b_1(a_2) = 0.5, b_2(a_1) = 0.25, b_2(a_2) = 0.75$.
In this simple example it was possible to find our solution by hand,. However real problems often have long sequences with many states, so such a complete calculation cannot be done even by computer.

4.2 Hidden Markov models and multiple sequence alignments

The following figure illustrates how a Hidden Markov Model can be used to model a protein family.

FIGURE 14.1

The example has length 5, although any length is possible. States are labelled m_j, i_j and d_j. m, i and d stand for *match, insert* and *delete*. We start in state m_0 and move from left to right ending on m_5, through some path denoted by the arrows.

Emissions are made from all states on the path through our diagram except the first (m_0) and last (m_5, here). Our alphabet consists of the twenty amino acids, plus a dummy symbol δ representing delete. Each match and insert state has its own distribution over the 20 amino acids and cannot emit δ, a delete state must emit δ.

It could be that all emissions in m and i are uniformly likely, which would mean that random sequences would result, or ms could have one distribution, and is another. It is possible to find the most likely path through the model by using an algorithm, such as the *Viterbi* algorithm.

For example, consider the sequences CAEFDDH and CDAEFPDDH and suppose that their most likely paths are
$m_0m_1m_2m_3m_4d_5d_6m_7m_8m_9m_{10}$ and
$m_0m_1i_1m_2m_3m_4d_5m_6m_7m_8m_9m_{10}$ respectively.

The alignment induced by this model is found by aligning positions generated by the same state, as follows:

FIGURE 14.2

This leads to the induced alignment
C – A E F – D D H
C D A E F P D D H

Consider an extension of this example, with more than two sequences. Suppose that we have sequences CAEFTPAVH, CKETTPADH, CAETPDDH, CAEFDDH, CDAEFPDDH and the corresponding paths are

$m_0m_1m_2m_3m_4m_5m_6m_7m_8m_9m_{10}$
$m_0m_1m_2m_3m_4m_6m_7m_8m_9m_{10}$
$m_0m_1m_2m_3d_4m_5m_6m_7m_8m_9m_{10}$
$m_0m_1m_2m_3m_4d_5d_6m_7m_8m_9m_{10}$
$m_0m_1i_1m_2m_3m_4d_5m_6m_7m_8m_9m_{10}$

The induced alignment is
C – A E F T P A V H
C – K E T T P A D H
C – A E – T P D D H
C – A E F – – D D H
C D A E F – P D D H

4.3 Bootstrapping methods

With the advent of fast powerful computers, it has become possible to use alternative methods to test statistical hypotheses and find confidence intervals.

We use the *empirical probability distribution* of a set of data. If we have a set of data values $x_1, \ldots, x_n$ then the empirical probability distribution of our random variable X is

$$P_e(X = x) = \frac{m_x}{n}$$

where m_x is the number of times x appears in our data set.

Question: What is the expectation of the value of the proportion of the data less than or equal to a given x (this proportion is the empirical distribution function at x)?

Suppose that we want to find an estimate of the mean of the underlying distribution and a confidence interval for this mean. one way to do this is given in Section 2 of the course. This procedure assumed that the sample mean $\bar{X}$ has a normal distribution (at least approximately). If this is not the case, then the methodology used is not valid. The aim of the bootstrap method is to provide a way of finding estimates and confidence intervals which do not need this assumption.

Suppose that we start with the data set $x_1, \ldots, x_n$. The first step of the bootstrap involves sampling from the this collection n times, *with replacement*. Some observations from the original data may not appear at all, others will appear once, others twice etc.

Example 14.2 - We consider the following simple example of $n = 12$ data points (these have actually been randomly generated from an exponential distribution with parameter 0.5, and so mean 2)

0.843, 0.977, 0.003, 3.159, 1.027, 1.009, 3.331, 0.235, 0.476, 0.398, 4.597, 0.434

Each of the 12 observations in each bootstrap sample is selected from the above 12 numbers, each being chosen at random with probability 1/12. The number of times the value 0.843 appears in a sample, for example, follows a binomial distribution with parameters (12, 1/12). On average it appears once (occurring with probability 0.384), but it may not appear at all (probability 0.352) or more than once (the probabilities that it appears 2 and 3 times are 0.192 and 0.058 respectively).

From this sample we replace $x_1, \ldots, x_n$ by the bootstrap sample values, to give $\bar{x_{B1}}$, the mean of the first bootstrap sample. The procedure is then repeated a large number R times, leading to R bootstrap estimates

$$\bar{x_{B1}}, \ldots, \bar{x_{BR}}$$

These can be thought of as giving an empirical distribution of $\bar{X}$. The average of the bootstrap estimates is the *bootstrap estimate* of the mean.

Example 14.2 continued - we shall now take 20 bootstrap samples from the above distribution (recall that we sample with replacement), giving the mean value for each sample. These means are as follows.

Means - 1.510, 1.685, 0.952, 0.961, 1.230, 0.446, 2.211, 1.921, 1.295, 1.669, 0.906, 0.825, 1.495, 1.462, 1.326, 1.261, 1.157, 1.194, 1.262, 1.425

A 95% confidence interval for the mean of the population can be obtained from the above, and is (0.659, 2.218).
Note that this contains the true mean that we know to be 2, due to the method that the data was generated.

Typically bootstrapping is useful when there is good reason to think that the samples that we take may be significantly non-normal. Thus we can find confidence intervals for a wide range of parameters that do not behave as nicely as means. Thus in the above example we may be interested in the variance of the data. With a small sample, the distribution of the sample variance is quite different to the normal distribution. There is an alternative way to carry out tests for the variance that we have seen before; but how about the mean of the reciprocal of the data $1/x_i$ for example. Any function $\theta(X_1, \ldots, X_n)$ can be treated in the same manner.

It is possible to use the bootstrap procedure with more than one sample. For instance we may be interested in finding a confidence interval for the difference of two means, but not be confident that the underlying variances of the distributions are equal, or that the data is sufficiently close to being normally distributed. The easiest way to do this is to generate R pairs of samples from each data set independently using the method above. If we have n data points from sample 1 and m from sample 2, each bootstrap sample (pair) also contains n and m data points, the mean of each being taken and the difference of the means being the summary statistic from the data. These differences for each sample can then be considered together to provide an estimate and confidence interval for the difference of the means. Note that there are also more elaborate ways to do this, which are preferred by some researchers.

5 Monte Carlo Markov Chains II

In this section we consider two of the most important methods of Monte Carlo Markov Chains, together with an example of their use which relates to earlier problems in sequence analysis. We shall show in each case the mathematical detail of how the method works.

5.1 The Metropolis-Hastings Algorithm

The aim of the Metropolis-Hastings Algorithm is to construct a Markov Chain which is both aperiodic and irreducible, which has a given stationary distribution.

Let us suppose that we have a stationary distribution $\mathbf{v}$, which is of course a vector of probabilities defined on some number of states n. To construct our Markov Chain, firstly we choose some set of constants $q_{ij} \quad i, j = 1, \ldots, n$
such that $q_{ij} > 0$ for all i, j and $\sum_j q_{ij} = 1$ for all i. We shall now define a_{ij} by

$$a_{ij} = min\left(1, \frac{v_j q_{ji}}{v_i q_{ij}}\right)$$

We define p_{ij} by

$$p_{ij} = q_{ij} a_{ij} \qquad i \neq j$$

and

$$p_{ii} = 1 - \sum_{j \neq i} p_{ij}$$

Since $q_{ij} > 0$ we can see that $p_{ij} > 0$ for all i, j pairs (including when $i = j$ although this is not as immediately obvious). Thus the Markov Chain defined by the transition probabilities is aperiodic and irreducible. It just remains to show that the stationary distribution of this Markov Chain is $\mathbf{v}$.

This is shown by demonstrating that $v_j p_{ji} = v_i p_{ij}$ for all i, j pairs. This is called the *detailed balance requirement*, and is a sufficient condition for $\mathbf{v}$ to be a stationary distribution of the process. Essentially, the rate of movement from state i to state j is balanced by the rate from state j to state i, and so the process is in (stochastic) equilibrium.

If we suppose that $v_j q_{ji} < v_i q_{ij}$, then from above we have

$$a_{ij} = \frac{v_j q_{ji}}{v_i q_{ij}}, p_{ij} = \frac{v_j q_{ji}}{v_i}, a_{ji} = 1, p_{ji} = q_{ji}$$

and it follows that $v_j p_{ji} = v_i p_{ij}$.
If the reverse inequality holds the reasoning is identical; if $v_j q_{ji} = v_i q_{ij}$ then it also follows easily.

This method can be extended to the case where some q_{ij} are zero (important for the following section) provided that whenever $q_{ij} > 0$, so is q_{ji}.

Why should we wish to do this? We may wish to take a random sample from this distribution, or to find the state with the largest probability from the distribution. But for any reasonably sized number of states, it is easier to compute these from the distribution directly rather than construct the Markov Chain and let it run through many iterations. The answer to this question is that sometimes the number of states is just so large, and the definition of the probabilities of being in the state given in a sufficiently indirect way, that this is impossible. The Markov chain idea can give us a shortcut round this problem, as we shall see in the next section.

5.2 Gibbs Sampling

Suppose that $Y_i, i = 1, \ldots, k$ are discrete finite random variables, and $\mathbf{Y}$ is the random column vector made up of the Y_is, i.e. $(Y_1, \ldots, Y_k)^T$. We can define the distribution of $\mathbf{Y}$, $P_{\mathbf{Y}}(\mathbf{y})$, the probability that $\mathbf{Y}$ takes the vale $\mathbf{y}$ (i.e. $Y_i = y_i$ for all i). We assume that $P_{\mathbf{Y}}(\mathbf{y}) > 0$ for all values of $\mathbf{y}$.

We will construct a Markov Chain whose states are the possible values of $\mathbf{Y}$. We shall order the vectors $\mathbf{y}$ in some manner, to give vectors numbered from 1 to n. We equate vector j with state j in our Markov chain.

We define the Markov chain as follows;
We consider each of the k components of the vector in turn, deciding on a new value for this component only in each step, which may be the same as the old one [alternatively we pick one of the k components at random and only perform a single step -see the next section] . We construct a transition matrix $P^{(1)}$ for the first step as follows. If vectors i and j differ in any component but our chosen one, then we set $p_{ij}^{(1)} = 0$.
If they differ by this first component only (or not at all) we define

$$p_{ij}^{(1)} = P(Y_1 = y_1^* | Y_2 = y_2, \ldots, Y_k = y_k) = \frac{P(Y_1 = y_1^*, Y_2 = y_2, \ldots, Y_k = y_k)}{P(Y_2 = y_2, \ldots, Y_k = y_k)}$$

where vector i is $(y_1, y_2, \ldots, y_k)$ and vector j is $(y_1^*, y_2, \ldots, y_k)$.

After we have changed the first component (or left it the same) we move on to the second component, performing the same operation above to obtain the transition matrix for this component $P^{(2)}$ and so on. A move in the full Gibbs process is a sequence of k moves, one for each component, and thus the full transition matrix for the Gibbs sampler is just the product of all of these transition matrices for each step

$$P = P^{(1)} P^{(2)} \ldots P^{(k)}$$

Exercise: Explore the simplest case of this idea, with a two-dimensional vector, each position having two elements only. This is the situation in exercise E2.

This Markov chain is irreducible and aperiodic, since $p_{ii} > 0$ for all i and every state can be reached by a finite number of steps from any other (it is possible to move between two vectors which differ in l places in l steps).

It also has stationary distribution $P_{\mathbf{Y}}(\mathbf{y})$. Choosing $q_{ij} = p_{ij}$, we can show that both $v_j q_{ji}$ and $v_i q_{ij}$ are equal and so $a_{ij} = 1$ and $p_{ij} = a_{ij} q_{ij}$, matching the previous section, confirming the above stationary distribution.

5.3 Gibbs Sampling for multiple sequence alignments

Suppose that we wish to compare more than two sequences to try and find the best alignment between them. If we have N sequences and they are each of approximate length L, then the number of (global) alignments is of the order of

$$(2L)^N$$

which can easily be far too large to handle in the conventional manners described in section 13. We shall use the Gibbs sampling method from the previous section to try to find good alignments (it is usually too ambitious to try to find the "best" alignment out of so many possibilities). We consider the example of protein sequences.

Label the amino acids in some order $1, \ldots, 20$. Supposing we now have N protein sequences of lengths $L_1, \ldots, L_N$, the aim is to find N segments, one form each sequence, each of some fixed length W which in some way are similar to each other. There are

$$\prod_{i=1}^{N} (L_i - W + 1)$$

possible choices for the locations of these N segments. We assume that N and the L_j are large enough that a simple algorithmic approach is not feasible. We shall consider each of these possible choices as a state in a Markov chain.

The procedure works as follows. We follow a series of simple steps, moving from one state of the Markov chain (i.e. one array) to another. The figures below illustrate this. In Figure 15.1 we choose the third sequence, so the third row is changed in step 1, and the first sequence (so the first row) in step 2.

FIGURE 15.1

In fact each step has two parts; selection of the row to be changed, and the manner of its change. Figure 15.2 shows the first part of the step only. Which row is chosen is a purely random choice [or we could pick each row in turn as described in the previous section]. The aim of the second part, is to improve the overall alignment, by choosing one of the $L_i - W$ alternative segments.

FIGURE 15.2

Suppose that in the first reduced array of Figure 15.2 (i.e. with row 3 removed) amino acid j occurs c_{ij} times in the ith column. From these values, we find a probability estimate

$$q_{ij} = \frac{c_{ij} + b_j}{N - 1 + B}$$

where the b_j are what are called pseudocounts. They can be chosen in a number of ways, but one sensible choice is $b_j = p_j$, the background frequency of amino acid j.

Suppose that the amino acid sequence in a given segment is $\mathbf{x}= x_1, x_2, \ldots, x_W$. The probability of this ordered set under the population amino acid frequencies is

$$P_x = p_{x1} p_{x2} \ldots p_{xW}$$

The estimated probability of this segment under the $N - 1$ segments in our reduced array is

$$Q_x = q_{1x1} q_{2x2} \ldots q_{WxW}$$

The likelihood ratio (which here is an estimate of how much more likely our sequence is due to similarity to the other sequences over pure chance, is $LR(x) = Q_x/P_x$. We select segment $\mathbf{x}$ with probability

$$\frac{LR(x)}{\sum_{m=1}^{L_i-W} LR(m)}$$

where the denominator is the sum of all the likelihood ratios of the $L_i - W$ possible segments.

As time goes on, we should see better and better alignments appearing more frequently.

The relative entropy between $q^*_{ij}(s)$ and p_j is

$$\sum_{i=1}^{W}\sum_{j=1}^{20} q^*_{ij}(s) log\left(\frac{q^*_{ij}(s)}{p_j}\right)$$

States for which the relative entropy is high are those which represent good alignments. Note that this entropy is approximately a linear function of the logarithm of the probability of being in state s

$$C\prod_{i=1}^{W}\prod_{j=1}^{20}\left(\frac{q^*_{ij}(s)}{p_j}\right)^{c_{ij}(s)}$$

This method is an example of the Gibbs sampling method that we described above. Each state of the process is a vector with N elements (one for each of our sequences) and there are a finite number of possible values for each element $L_i - W + 1$ for element i.

FIGURE 15.3

Each step of the process changes a single sequence only (i.e. changes a single element of the vector) so that $p_{ij} = 0$ if elements i and j differ by more than a single element. Since the probability expression

$$P(Y_1 = y_1, Y_2 = y_2, \ldots, Y_k = y_k)$$

is almost identical to the likelihood ratio $LR(x)$ with the appropriate probabilities defined above, it follows that this is essentially an example of a Gibbs sampling procedure.

6 Genetics

6.1 Hardy-Weinberg equilibrium

The Hardy-Weinberg principle states that, under certain conditions, after a single generation of random mating, the genotype frequencies at a single locus will be fixed at a particular equilibrium value, which is the *Hardy-Weinberg equilibrium.*

Suppose that there are two alleles at a given locus, labelled by A and a, and the proportions of these in the population are p and $q = 1 - p$ respectively. A population is in Hardy-Weinberg equilibrium if the proportion of zygote AA in the population is p^2 and the proportion of the heterozygote Aa is $2pq$. These proportions are those that would be obtained if the genes were selected from the population at random; the number of As at the locus is simply Binomially distributed, with parameters 2 and p.

Exercise: Show that after a generation of random mating a population with two alleles is in Hardy-Weinberg equilibrium, irrespective of the original distribution of zygotes.

The original assumptions for a population to be in Hardy-Weinberg equilibrium were that the population is; diploid, sexually reproducing and randomly mating. in addition the population does not suffer drift, selection, mutation or migration.

The principle can easily be generalised to the case with multiple alleles. Suppose that there are n alleles at a locus

$$A_1, \ldots, A_k$$

and these have population frequencies of $A_1, \ldots, A_k$ respectively, then under Hardy-Weinberg equilibrium the frequency of homozygote A_iA_i is p_i^2 and the frequency of heterozygote A_iA_j is $2p_ip_j$ for all values of i, j.

We may be interested in whether a particular population is in Hardy-Weinberg equilibrium, or if not, how much (in some sense) that it deviates from it. We shall explore different methods of testing for this.

6.2 The chi-square test

One way of testing for hardy-Weinberg equilibrium is to use the chi-square test that we saw last term in the general context of tabular data. Below we reproduce a revised version of the table from Section 7. In this case we have the entry Y_{ij} $i \leq j$ representing the number of individuals with genotype A_iA_j. The table has the same number of rows and columns (equal to the total number of alleles k). Since A_iA_j and A_jA_i are the same, we only need a single category, and we choose the row i column j entry for $i < j$. The total number of individuals is thus

$$N = \sum_{i \le j} Y_{ij}$$

$$\begin{array}{cccc}
Y_{11} & Y_{12} & \ldots & Y_{1k} \\
Y_{21} & Y_{22} & \ldots & Y_{2k} \\
- & - & \ldots & \ldots \\
- & - & - & Y_{kk}
\end{array}$$

The null hypothesis is that the population is in Hardy-Weinberg equilibrium.

Under H_0 the statistic is

$$Q = \sum_{i=1}^{k} \sum_{j=i}^{k} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

where $O_{ij} = Y_{ij}$ is the observed number of individuals with genotype $A_i A_j$ and E_{ij} is the expected number of observations in this cell. E_{ij} is found simply by multiplying the total number of individuals by the probability that a random individual (under Hardy-Weinberg equilibrium) has that genotype. Hence if we knew the values of p_is we would obtain

$$E_{ii} = p_i^2 N \quad E_{ij} = 2 p_i p_j N \quad i < j$$

Usually we have to estimate the p_is from the data. We estimate p_i (the proportion of allele A_i in the population) by the proportion of allele A_i in the sample, which is

$$\hat{p_i} = \frac{\sum_{i<j} Y_{ij} + \sum_{i>j} Y_{ji} + 2Y_{ii}}{2N}$$

Q has approximate distribution χ^2 with $k(k-1)/2 + k - k = k(k-1)/2$ degrees of freedom (the number of cells minus the number of terms estimated from the data).

Example 16.1 - suppose that we have a system with two alleles where the zygotes $A_1 A_1$, $A_1 A_2$ and $A_2 A_2$ occur in frequencies 19, 52, 27 respectively. In the above notation, this gives

$$Y_{11} = 19, Y_{12} = 52, Y_{22} = 27, N = 98$$

The proportion of allele A_1 in our data sample is

$$\frac{52 + 2 \times 19}{2 \times 98} = 0.4592$$

and so the proportion of A_2 is 1-0.4592=0.5408. Thus our population proportion estimates are $\hat{p_1} = 0.4592$ and $\hat{p_2} = 0.5408$.

Under the null hypothesis of Hardy-Weinberg equilibrium, the expected frequencies of the three zygotes are
$E_{11} = 98(0.4592)^2 = 20.66$
$E_{12} = 2 \times 98(0.4592)(0.5408) = 48.67$
$E_{22} = 98(0.5408)^2 = 28.66$

$$Q = \sum_{i=1}^{k} \sum_{j=i}^{k} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} = \frac{(19 - 20.66)^2}{20.66} + \frac{(27 - 28.66)^2}{28.66} + \frac{(52 - 48.67)^2}{48.67} = 0.457$$

This should be χ_1^2. The 95% point of this distribution is $3.84 > 0.457$. So we do not reject the null hypothesis that the population is in Hardy-Weinberg equilibrium.

6.3 Fisher's exact test

Fisher's exact test is a version of the Likelihood Ratio Test, useful especially when some categories contain small numbers. The likelihood function when Hardy-Weinberg equilibrium is not assumed is

$$\prod_{i=1, j \geq i} q_{ij}^{Y_{ij}}$$

where there are no restrictions on the values of q_{ij}, except that they all add to 1. When there are 2 alleles, this becomes

$$q_{11}^{Y_{11}} q_{12}^{Y_{12}} q_{22}^{Y_{22}}$$

Under the null hypothesis of Hardy-Weinberg equilibrium, $q_{ij}, i \neq j$ is replaced by $2p_i p_j$ and q_{ii} by p_i^2 and for two alleles the likelihood function becomes

$$p^{2Y_{11}} (2p(1-p))^{Y_{12}} (1-p)^{2Y_{22}}$$

where p is the frequency of A_1 in the population.

We proceed to find the maximum likelihood function under the null hypothesis, and without restrictions. The ratio of these two is the likelihood ratio Λ.

It can be shown that with no restriction, the maximum likelihood estimates (the third is implied by the other two, since all probabilities must add to 1) are

$$\hat{q_{11}} = \frac{Y_{11}}{N}, \hat{q_{12}} = \frac{Y_{12}}{N}, \hat{q_{22}} = \frac{Y_{22}}{N}$$

and so the likelihood function is

$$\left(\frac{Y_{11}}{N}\right)^{Y_{11}} \left(\frac{Y_{12}}{N}\right)^{Y_{12}} \left(\frac{Y_{22}}{N}\right)^{Y_{22}}$$

Under the null hypothesis the maximum likelihood estimate of p is

$$\hat{p} = \frac{Y_{11} + Y_{12}/2}{N}$$

and the corresponding likelihood function is

$$\left(\frac{Y_{11} + Y_{12}/2}{N}\right)^{2Y_{11}+Y_{12}} \left(\frac{Y_{22} + Y_{12}/2}{N}\right)^{2Y_{22}+Y_{12}} 2^{Y_{12}}$$

$-2log(\Lambda)$ can be shown to be

$$2(Y_{11}log(Y_{11}) + Y_{12}log(Y_{12}/2) + Y_{22}log(Y_{22}) + Nlog(N)$$

$$-(2Y_{11} + Y_{12})log(Y_{11} + Y_{12}/2) - (2Y_{22} + Y_{12})log(Y_{22} + Y_{12}/2))$$

Under the hypothesis of Hardy-Weinberg equilibrium, this quantity has an approximate χ_1^2 distribution.

Considering the above example with $Y_{11} = 19, Y_{12} = 52, Y_{22} = 27, N = 98$

$-2log(\Lambda) = (19log(19)+52log(26)+27log(27)+98log(98)-90log(45)-106log(53)) = 0.458$

$0.458 < 3.84$ similarly to above, so we do not reject the null hypothesis of Hardy-Weinberg equilibrium.

6.4 Estimating the Hardy-Weinberg disequilibrium

We have seen above how to test to see whether Hardy-Weinberg equilibrium is plausible for a given set of data. What if it is not (or you think that it may well not be true) so that you wish to estimate the degree of disequilibrium? An estimate can be found by comparing the true proportion of the heterozygote AB, with the predicted level if it was under Hardy-Weinberg. The disequilibrium is defined as half of this difference

$$D = \frac{1}{2}\left(\frac{Y_{12}}{N} - 2\hat{p}(1-\hat{p})\right)$$

We can find an approximate confidence interval by using the Hardy-Weinberg proportions to estimate the variance of $Y_{12}/2N$. This estimate is

$$\frac{\hat{p}(1-\hat{p})}{4N}$$

So that the 95% confidence interval is

$$D - 1.96\sqrt{\frac{\hat{p}(1-\hat{p})}{4N}}, D + 1.96\sqrt{\frac{\hat{p}(1-\hat{p})}{4N}}$$

Thus in our example above,

$$D = \frac{1}{2}\left(\frac{52}{98} - 2 \times 0.4592(1-0.4592)\right) = 0.0170$$

and the 95% confidence interval is

$$D - 1.96\sqrt{\frac{0.4592(1-0.4592)}{4 \times 98}}, D + 1.96\sqrt{\frac{0.4592(1-0.4592)}{4 \times 98}} = (-0.0323, 0.0663)$$

7 Phylogenetic trees and evolutionary models I

7.1 Introduction to phylogenetic trees

The evolutionary relationships between a set of species with a common ancestor can be represented by a binary tree, as we see in the following figure. Species are represented by points (nodes) connected by lines (edges).

FIGURE 17.1

The lengths of the edges represent evolutionary time, the longer an edge, the more time separates the nodes at the ends of the edge. The trees in Figure 17.1 are examples of rooted trees, where there is a single root node representing the common ancestor. An unrooted tree indicates the relationship between species without showing the direction of evolutionary time, as in Figure 17.2.

FIGURE 17.2

For a set of species there will be some real, unknown, phylogenetic tree connecting the members. Our aim is in infer what this tree is as accurately as possible, given the data. We shall look at several methods of doing this.

7.2 Distances

Several methods and algorithms are based on the concept of a *distance* between species. The most natural measure of distance between two species is the number of years since their most recent common ancestor. This is usually unknown, so in practice we have to use a surrogate distance in its place.

let us assume that exact distances are known. For a set of points S, any measure of distance d must satisfy the following for all elements x, y, z in S.

$$(i) d(x,y) \geq 0$$

$$(ii) d(x,y) = d(y,x)$$

$$(iii) d(x,y) \leq d(x,z) + d(z,y)$$

A distance is a *tree-derived* distance if there is a tree with these species at the leaves such that the distance between x and y is the sum of the lengths of the edges joining them. This automatically satisfies $(i) - (iii)$ above. In fact it turns out that (iii) becomes

$$(iii) d(x,y) < d(x,z) + d(z,y)$$

for such a tree.

For rooted trees joining extant species (non-extinct, so being measured at identical times) there are further restrictions. For the tree in Figure 17.3 the conditions
$d(x, y) = d(x, z)$
and $d(y, z) < d(x, y), d(y, z) < d(x, z)$
must hold.

FIGURE 17.3

Any distance measure which satisfies the condition that for any three members x, y, z of S, two of the three distances between them are equal and the third is smaller than these two, as above, is called *ultrametric*. So a rooted tree of extant species with a tree-derived distance is always ultrametric.

7.3 Tree reconstruction: the ulttrametric case

We shall give an outline of how to show that if an ultrametric distance measure is given for all species then there is a unique rooted tree joining the species (except for trivial changes).

Suppose that we have a set of species $s_1, s_2, \ldots, s_n$ with an ultrametric distance measure $d(x, y)$ between each pair of species x and y.

We shall use an induction argument to show the existence of a unique tree. For only two species, then both must be the same distance from the root and we obtain the following tree.

FIGURE 17.4

Suppose that we can find a unique tree (except for trivial changes like swapping s_1 and s_2 in the above) for m species. If we can show that under this assumption we can do the same for $m + 1$ then we can do it for any number (we can find a tree for 2, and so for 2+1=3, and so for 3+1=4 etc.). For our m species, there is a single root, with a collection of species (labelled S_L) down the left hand branch, and another collection (S_R) down the right-hand one. Let x be some element of S_L and y some element of S_R.

FIGURE 17.5

Now consider the extra species s_{m+1} and its distance from x and y. The ultrametric property indicates that one of x, y and s_{m+1} is equidistant from the other 2. Thus there are three possibilities:

$$(1) d(s_{m+1}, x) = d(s_{m+1}, y) > d(x, y)$$

leads to the following tree (where $a = d(s_{m+1}, x)/2$ and $b = (d(s_{m+1}, x) - d(x, y))/2)$

FIGURE 17.6

We thus have the correct distances between x, y and s_{m+1}. It is not hard to show that this must also work for every other species as well.

$$(2) d(s_{m+1}, y) = d(x, y) > d(s_{m+1}, x)$$

This puts the species s_{m+1} on the same side of the root as S_L. If s_{m+1} is equidistant from every member of S_L we get the following tree

FIGURE 17.7

If not, s_{m+1} is closer to some members of S_L than others, and we can follow similar arguments within S_L to find its correct place.

The third case is

$$(3) d(s_{m+1}, x) = d(x, y) > d(s_{m+1}, y)$$

Case (3) is the same as case (2), just with S_R and S_L interchanged.

7.4 Tree reconstruction: neighbour joining

Two species are *neighbours* if the path between them contains only one node. Thus in the following tree, x and y are neighbours, but x and z are not.

FIGURE 17.8

The following describes the UPMGA (unweighted pair group method using arithmetic averages) algorithm. The algorithm describes distances between groups of species, starting with 'groups' of a single species.

We define the distance between groups u and v as

$$d^*(G_u, G_v) = \frac{1}{n_1 n_2} \sum_{x \in G_u, y \in G_y} d(x, y)$$

where n_1 and n_2 are the sizes of groups G_u and G_v. We find the smallest distance between two groups, and join them with a two-leaf rooted tree, with root "species" r_1 as follows

FIGURE 17.9

The distance from any node z to this root is given by;

$$d(r_1, z) = \frac{1}{2}(d(x, z) + d(y, z) - d(x, y))$$

We now replace the two groups G_r and G_s by the group G_{rs} containing the elements of both groups. We repeat the above procedure, now with one group less, until eventually we have a complete tree.

The *neighbour-joining* algorithm is a more complex procedure, based upon the function

$$\delta(x, y) = (N - 4)d(x, y) - \sum_{z \neq x,y} (d(x, z) + d(y, z))$$

It can be shown that this reaches its minimum value if and only if x and y are neighbours, so that the algorithm joins such pairs immediately. Distances from the node (r_1) joining any such pairs are now calculated, again using;

$$d(r_1, z) = \frac{1}{2}(d(x, z) + d(y, z) - d(x, y))$$

Exercise: Repeat E3 using the neighbour-joining algorithm, instead of the UPGMA.

7.5 Surrogate distances

Unfortunately it is rarely the case that exact distances between species are known, and we must use surrogate distances instead. These are usually derived using DNA information from the species being considered. Evolutionary changes can happen faster along one branch than another for a number of reasons, for example different generation lengths, so we may obtain a tree like Figure 17.10.

FIGURE 17.10

In this example the molecular clock ran slower for the elephant than the others. A surrogate distance is often found by using aligned DNA sequences taken from two species, where the distance is proportional to

$$-log\left(1 - \frac{4}{3}p\right)$$

where p is the proportion of nucleotides where the two sequences differ. This estimate derives from the Jukes-Cantor model, which we shall meet in the next section. We can then use the methods that we have already described to estimate the tree with the surrogate distances instead of the distances (note that the basic properties of a distance (i)-(iii) may not be satisfied, so that negative distances can appear in the inferred trees). Quite often more complex versions of this type of idea need to be used.

7.6 Tree reconstruction: parsimony

Using the method of parsimony, a cost is assigned to each tree, and the optimal tree is the one which minimises the cost. The aim is to find the optimal tree *topology.*

Consider the cost function where unit cost is made for each nucleotide substitution. We find the optimal tree in two steps. Firstly all possible tree topologies must be listed, with species allocated to the leaves. Secondly, for each such choice, the labelling of all internal nodes with a suitable DNA sequence, to ensure minimisation of the cost must be found (this is done using *Fitch's algorithm*).

For a large number of species, the first step is the hardest. For three species there is only one possible topology, with three labelling choices. The following figure shows the five optimal allocations of internal nodes for the simplistic sequences AA, AB and BB.

FIGURE 17.11

The number of topologies increases rapidly with species number. For example, for 20 species there are 8×10^{21} topologies. It is possible to use versions of the above methodology on very large numbers of species. The distinctive feature of this method is that it does not construct distances between species, only the topology.

8 Phylogenetic trees and evolutionary models II

8.1 Introduction to evolutionary models

We shall introduce some models of the evolution of biological data which will be useful for the construction of phlyogenetic trees. We shall assume that there is one dominant nucleotide at any given site for a particular species, so that each species has its own "genome". Over time the nucletides at any site may change; such changes are assumed to occur over negligible times (in evolutionary time this is reasonable). Such a change is called a *substitution.*

8.2 The Jukes-Cantor model

The simplest model of nucleotide substitution is the jukes-Cantor model. The discrete time version considers a Markov chain with four states a, g, c, t with a transition matrix (probability of moving from one state to another in unit time) given by

$$P = \begin{vmatrix} 1-3\alpha & \alpha & \alpha & \alpha \\ \alpha & 1-3\alpha & \alpha & \alpha \\ \alpha & \alpha & 1-3\alpha & \alpha \\ \alpha & \alpha & \alpha & 1-3\alpha \end{vmatrix} \tag{8}$$

The value of α will obviously depend upon the chosen timescale. Thus whatever nulceotide exists at whichever site, it is equally likely to be substituted by any of the other three.

The stationary distribution of this Markov chain is just
$\phi = (0.25, 0.25, 0.25, 0.25)$.
It is possible to show that after n steps the transition matrix P^n is

$$P^n = \begin{vmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{vmatrix} \tag{9}$$

$$+(1-4\alpha)^n \begin{vmatrix} 0.75 & -0.25 & -0.25 & -0.25 \\ -0.25 & 0.75 & -0.25 & -0.25 \\ -0.25 & -0.25 & 0.75 & -0.25 \\ -0.25 & -0.25 & -0.25 & 0.75 \end{vmatrix} \tag{10}$$

Thus, whatever the dominant nucleotide at time 0, the probability that it is still the dominant one at time n is

$$0.25 + 0.75(1 - 4\alpha)^n$$

and the probability that it is any specific other one is

$$0.25 - 0.25(1 - 4\alpha)^n$$

8.3 The Kimura models

The assumption that all transitions between nucleotides are equally likely is unrealistic, and some are more likely than others. In particular, transitions between a and g, and between c and t are more common than the others. Kimura's first model, allows for two different trnasition probabilities to occur; α is the probability of moving between a and g, or c and t and other changes have rate β. Thus the transition matrix becomes

$$P = \begin{vmatrix} 1-\alpha-2\beta & \alpha & \beta & \beta \\ \alpha & 1-\alpha-2\beta & \beta & \beta \\ \beta & \beta & 1-\alpha-2\beta & \alpha \\ \beta & \beta & \alpha & 1-\alpha-2\beta \end{vmatrix} \tag{11}$$

The stationary distribution of this Markov chain can again be shown to be the simple equal probability one,
$\phi = (0.25, 0.25, 0.25, 0.25)$.
As in the Jukes-Cantor model, we can find the transition matrix P^n.

$$P^n = \begin{vmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{vmatrix} \tag{12}$$

$$+(1-4\beta)^n \begin{vmatrix} 0.25 & 0.25 & -0.25 & -0.25 \\ 0.25 & 0.25 & -0.25 & -0.25 \\ -0.25 & -0.25 & 0.25 & 0.25 \\ -0.25 & -0.25 & 0.25 & 0.25 \end{vmatrix} \tag{13}$$

$$+(1-2(\alpha+\beta))^n \begin{vmatrix} 0.5 & -0.5 & 0 & 0 \\ -0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & -0.5 \\ 0 & 0 & -0.5 & 0.5 \end{vmatrix} \tag{14}$$

Thus, whatever the dominant nucleotide at time 0, the probability that it is still the dominant one at time n is

$$0.25 + 0.25(1 - 4\beta)^n + 0.5(1 - 2(\alpha + \beta))^n$$

The probability that it has been replaced by its ”partner” (e.g. a by g) is

$$0.25 + 0.25(1 - 4\beta)^n - 0.5(1 - 2(\alpha + \beta))^n$$

and the probability that it is a specific one of the other two is

$$0.25 - 0.25(1 - 4\beta)^n$$

There are other more complex Kimura models where each of the substitution probabilities differ, but we shall not look at those here. The Felsenstein Model and the HKY Model are two further generalisations on the same theme using Markov transition matrices for nucleotide substitutions.

The Felsenstein model generalises the Jukes-Cantor model in a different way. It recognises that the equilibrium distribution of nulceotides is often not uniform i.e. (0.25,0.25,0.25,0.25) and so gives a transition matrix which gives the equilibrium distribution $(\phi_a, \phi_g, \phi_c, \phi_t)$ for any such values. This matrix is

$$P = \begin{vmatrix} 1 - u + u\phi_a & u\phi_g & u\phi_c & u\phi_t \\ u\phi_a & 1 - u + u\phi_g & u\phi_c & u\phi_t \\ u\phi_a & u\phi_g & 1 - u + u\phi_c & u\phi_t \\ u\phi_a & u\phi_g & u\phi_c & 1 - u + u\phi_t \end{vmatrix} \tag{15}$$

It is not too hard to verify that the stationary distribution of this matrix is indeed $(\phi_a, \phi_g, \phi_c, \phi_t)$

The HKY Model is a further generalisation, with features from both the Kimura and Felsenstein models (these are both special cases of it). The transition matrix is given by

$$P = \begin{vmatrix} 1 - u\phi_g - v\phi_c - v\phi_t & u\phi_g & v\phi_c & v\phi_t \\ u\phi_a & 1 - u\phi_a - v\phi_c - v\phi_t & v\phi_c & v\phi_t \\ v\phi_a & v\phi_g & 1 - u\phi_t - v\phi_a - v\phi_g & u\phi_t \\ v\phi_a & v\phi_g & u\phi_c & 1 - u\phi_c - v\phi_a - v\phi_g \end{vmatrix} \tag{16}$$

Exercise: Show that the HKY model has the required stationary distribution.

In addition, each of these models can be converted to continuous time models quite easily, with transition probabilities being replaced by transition rates.

8.4 Tree reconstruction: maximum likelihood

Suppose that we already have the topology of the tree, and we are interested in estimating the various lengths of the edges. Finding a likelihood function is relatively straightforward, given an evolutionary model. Thus we choose our model (for instance the Jukes-Cantor model). As an example, suppose that we have data from 5 species, which are arranged in the following tree.

FIGURE 18.1

At a particular collection of sites the nucleotide present for each of the species is known (and we assume that this is all that is known). We shall follow one such site, and label the nucleotides of the five species as A_1, A_2, A_3, A_4 and A_5 (some of these can of course be the same). In the tree there are nine nodes, and our likelihood function will be a product of 9 terms. Firstly the probability that W is the nucleotide at node n_0 is written as ϕ_W (taken from the its overall frequency in the population). There are eight arms leading from one node to another; each has an associated probability of a change from the nucleotide at the start node to the one at the end in the time given by the length of the node. For example, the probability that a change from nucleotide X at node n_1 to A_3 at node s_3 is labelled $P_{XA3}(l_4)$. If the nucleotides at the internal nodes n_1, n_2 and n_3 are X, Y and Z respectively, then the likelihood for our arm lengths AND these nucleotides at the root and internal nodes is given by

$$\phi_W P_{WX}(l_1) P_{WZ}(l_2) P_{XY}(l_3) P_{XA3}(l_4) P_{YA2}(l_5) P_{YA1}(l_6) P_{ZA4}(l_7) P_{ZA5}(l_8)$$

We now compute the above formula for the $4^4 = 256$ possible combinations of nucleotides at the root and the internal nodes, and the sum of all of these terms is the likelihood of the arm lengths $l_1, l_2, \ldots, l_8$.

This procedure is now repeated over all the sites that we possess information on, and a full likelihood function is found, which is just the product of all the likelihood functions at the single sites.

We can, in principle, repeat this for all tree topologies, and then find the largest of these likelihood functions, which will be from the tree which becomes our maximum likelihood estimator. Of course, as with the other methods, when the number of species is large, this can be very complex.

Felsenstein suggested a plan of building a tree (unrooted) by starting with two species and adding other species successively. If at any stage there are $k-1$ species, it turns out that there are $2k-5$ ways of adding a further species. Each of these should be tried in turn and that with the maximum likelihood accepted. This does not necessarily lead to the tree at the end of the process with maximum likelihood, and different trees can be obtained with different starting species. Different orders can be tried, and the final result with the biggest likelihood accepted.

It has been noticed that for real data sets, simplified evolutionary models like those that we have discussed can give rise to severe errors in the estimation of branch lengths. Increasingly complex evolutionary models (for instance Kimura's model is more complex than the Jukes-Cantor one, which is a special case of it) improve matters somewhat, but significant errors can still occur . Thus sometimes it is necessary to work with the more complex models.

The following example is taken from Ewens and Grant and illustrates the modelling of the evolutionary relationship between 14 different mammals. These are as follows:
Marsupial Mole, Wombat, Rodent, Elephant Shrew, Elephant, Whale, Dolphin, Pig, Horse, Bat, Insectivore, Human, Sea Cow, Hyrax.
The sequences for the 14 species are in Ewens and Grant, page 405. The distances between the species were found using the Kimura model of the previous section, and are given in the following table.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 ma. mole	0	.11	.41	.44	.39	.37	.40	.37	.41	.36	.40	.37	.42	.39
2 wombat	.11	0	.39	.40	.36	.33	.35	.33	.35	.32	.33	.33	. 38	.34
3 rodent	.41	.39	0	.33	.30	.24	.25	.22	.28	.23	.25	.23	.32	.31
4 el. shrew	.44	.40	.33	0	.20	.26	.26	.25	.28	.28	.28	.26	.20	.21
5 elephant	.39	.36	.30	.20	0	.22	.23	.22	.25	.25	.24	.21	.11	.12
6 whale	.37	. 33	. 24	. 26	. 22	0	.03	.10	.16	.17	.18	.17	.22	.24
7 dolphin	.40	.35	.25	.26	.23	.03	0	.11	.16	.19	.18	.17	.22	.25
8 pig	.37	.33	.22	.25	.22	.10	.11	0	.17	.18	.19	.17	.24	.24
9 horse	.41	.35	.28	.28	.25	.16	.16	.17	0	.17	.21	.20	.25	.26
10 bat	.36	.32	.23	.28	.25	.17	.19	.18	.17	0	.15	.20	.27	.27
11 insectiv.	.40	.33	.25	.28	.24	.18	.18	.19	.21	.15	0	.19	.26	.26
12 human	.37	.33	.23	.26	.21	.17	.17	.17	.20	.20	.19	0	.22	.23
13 sea cow	.42	.38	.32	.20	.11	.22	.22	.24	.25	.27	.26	.22	0	.14
14 hyrax	.39	.34	.31	.21	.12	.24	.25	.24	.26	.27	.26	.23	.14	0

Four different methods were used to find the optimal phylogenetic tree using the distances from the above table. The methods were UPGMA, neighbour-joining, parsimony and maximum likelihood. The trees found using these methods are shown in Figure 18.1.

FIGURE 18.2

Some features are common to the trees formed by all four methods, whereas others vary from tree to tree. Thus the relationship between horse, pig, whale and dolphin is the same for all four trees. There are large differences in the placement of humans, however. In the UPGMA tree, ‘human’ is grouped with horse,pig, whale and dolphin; in the neighbour joining tree it is not that closely linked to those types. The parsimony tree puts humans in the same group as elephant, sea cow, hyrax and elephant shrew. The maximum likelihood tree places humans and rodents together.

9 Evolutionary and Genetic Algorithms

Evolutionary algorithms are stochastic searching models which are based upon the processes of evolution. They may be used to model actual evolutionary processes, and this is what we will assume in this section, but also to model completely different situations. For instance the classical travelling salesman problem can be approached using evolutionary algorithms.

Genetic algorithms are a special class of evolutionary algorithms and we will use the terms interchangeably from now on. The basic structure of a genetic algorithm is as follows. There will be a set of optimisation criteria in place to determine whether we have reached a suitable solution to our problem (or evolved to a stable population).

FIGURE 19.1

Step 1: Generate an initial population of individuals. This will usually be done completely randomly, with no thought given to the 'fitness' of individuals.

Step 2: Are the optimisation criteria met? If so we stop the process. This is extremely unlikely to happen at the start of the process. If they are not met we continue.

Step 3: Generate a new population. This is done in a number of steps. Not all have to be included in any particular process, but all are needed to replicate a real genetic population (although leaving some out may lead to broadly similar conclusions).

Step 3a: Selection occurs according to the fitness of the individual to decide which individuals reproduce.

Step 3b: Recombination. Parents are recombined to produce offspring.

Step 3: Mutation. all offspring will be mutated with a certain probability.

We can then compute the fitness of the offspring and replace the parents with the offspring as the members of the population.

Step 4: Are the optimisation criteria met? If so the process stops, as in step 2. if they are not, then step 3 is repeated with the 'new' population replacing the original at the start, and undergoing further selection, recombination and mutation.

Evolutionary algorithms thus operate in a similar way to the more traditional optimisation methods, but there are differences. The major differences are that evolutionary algorithms are probabilistic rather than deterministic (and information such as derivatives are not needed) and that at any point the process is not at a point but at a set of points (a population).

We will now consider the main parts of the process of the evolutionary algorithm in turn.

9.1 Selection

The selection step chooses which individuals generate new offspring in the next generation and is the most fundamental step in the process. To perform selection, we must have a measure of fitness of each individual. Each individual in the population has a reproduction probability based upon their own objective fitness value, and on that of all the other individuals in the population.

Thus the probability of any new individual being a copy of type i is $P_i(f_1, f_2, \ldots, f_n)$ for some function P_i such that

$$\sum_{i=1}^{n} P_i(f_1, f_2, \ldots, f_n) = 1$$

We talk in terms such as *selective pressure* the relative probability of the best individual being chosen compared to the average probability of selection, and *loss of diversity* which is the proportion of individuals in a population that are not selected at all (and so have no offspring) by the above method.

The simplest form of selection is what is known as *roulette wheel selection* or *stochastic sampling with replacement.* If there are n individuals in our current population with fitnesses $f_1, f_2, \ldots, f_n$ as above, and we wish to generate m new individuals for our *mating population* (i.e. that prior to recombination) in our next population we choose each of these m independently where each of a copy of individual i with probability

$$P_i(\mathbf{f}) = \frac{f_i}{f_1 + f_2 + \ldots + f_n}$$

Thus the number of individuals in the mating population who are copies of individual i follows a Binomial distribution with parameters m and $P_i(\mathbf{f})$.

There are many other types of selection; for example individuals can be placed spatially in an environment and selection occurs within local groups, so that characteristics can vary considerably between localities.

9.2 Recombination

The selection process has generated a mating population. Individuals are now paired according to some rule; completely at random, according to whether an individual has been designated male or female (and then at random) or within the locality for example.

When a pair is chosen, they have to generate offspring.

Let us suppose that individual 1 is described by the the sequence (we shall refer to this as the genome of the individual)

$$t_{11}, t_{12}, \dots, t_{1k}$$

and individual 2 by

$$t_{21}, t_{22}, \dots, t_{2k}$$

(for the purposes of our process k may be 1 or 2 or much more, typically each value representing an allele of some gene).

This recombination (random recombination) can be done simply by choosing the first element from the sequence for the offspring to be t_{11} or t_{21} each with probability 0.5, and similarly for each other value, all independently of each other. So our new individual is generated as in the following figure.

FIGURE 19.2

When our elements are continuous values, we can generate new elements by using a linear combination of the parental values, so that the ith element is

$$\alpha_i t_{1i} + (1 - \alpha_i)(1 - t_{2i})$$

where α_i is an observation from a uniform (0,1) distribution. There are other variants on this idea.

An alternative method is the *crossover* method, which takes into account the correlation between genes on the same chromosome. For instance, the single-point crossover chooses a value at random from the list of positions $0, 1, \dots, k$ and then creates two new individuals. If the chosen value is j, then our two new individuals would be

$$t_{11}, t_{12}, \dots, t_{1j}, t_{2j+1}, \dots, t_{2k}$$

and

$$t_{21}, t_{22}, \dots, t_{2j}, , t_{1j+1}, \dots, t_{1k}$$

There are, again, many variants on this idea.

9.3 Mutation

We thus have a population of new individuals who are perfect copies of a combination of their two parents. There will be many genomes which can never be reached by selection and recombination alone; and they might be the fittest. Mutations prove very useful in helping to find the best genomes.

Mutations are random alterations in individuals. The mutation rate is related to the probability of a change in any given element of the genome of the individual. Generally to obtain good results, this mutation rate should decrease with the number of variables in the genome of our individuals; if there are n elements in the genome, a mutation rate of $1/n$ is sensible.

How mutations should occur, in the sense of which allele should be generated when a mutation occurs from a given allele, depends upon the form of alleles. If they are continuous values, or form a sequence of discrete values where there is an obvious ordering, then generally mutations should be small in size most of the time, but occasionally of large size.

For instance for continuous data, the mutation distance can be normally distributed, possibly centred on the original value (but possibly not). Thus if we start at allele t, then there is no mutation with probability $1-r$ and a mutation occurs with probability r which sends t to s where the value of s follows a normal distribution with mean t and variance σ^2, where σ^2 is a measure of the size of mutations, the larger σ^2, the larger mutation distances tend to be.

9.4 Fitness values

After a cycle of selection, recombination and mutation, we finally have a new generation. At the start of the section on selection, we talked about the fitnesses f_i of individuals. These are in reality properties of the genome of the individual and the should be a general function of the genome which is its fitness. It could be something simple like the sum of the genetic components $t_{11} + t_{12} + \ldots + t_{1k}$, for our first individual from before, so the larger the values of each t_{1i} the better, but will usually be something more complicated which it is difficult to see how to optimise.

The fitness of real genetic situations is of course very complex, but this is also true for the other types of problem that the genetic algorithm can be applied to, such as the travelling salesman problem.

The process continues until we find a satisfactory stopping point (perhaps the difference between the fittest and least fit individual is so small that all diversity is lost, barring mutations). The nature of the process means that this will generally occur at a point of high population fitness (but of course not necessarily the highest possible such point).

10 Dynamic programming

10.1 Introduction

Dynamic programming is a method of solving problems that are apparently complex but highly structured, by making use of the structure. The general approach is to break the problem down into subproblems and build the final solution using the solutions to each of the subproblems.

This type of approach is ideal for computers as it typically involves the repetition of many fairly straightforward calculations. One application of this idea is in games where there are sequential decisions to be made and where it is possible to identify all of the end-points of the game. The best strategy at each point depends upon what the best reply is; the game can be solved by working backwards from all of the possible final positions (backwards induction). *Noughts and crosses* can be completely solved in this way, but theoretically so could far more complex games like chess, if computers were powerful enough.

We shall consider one example of this approach that we met briefly in Section 13, which deals with the alignment of two DNA or protein sequences.

10.2 The Needleman Wunsch algorithm

The input for the Needleman Wunsch algorithm consists of two sequences

$$x = X_1 X_2 \dots X_m, \qquad y = Y_1 Y_2 \dots Y_n$$

of lengths m and n respectively. The elements of these sequences belong to some given alphabet of N symbols ($N = 4$ for DNA sequences and $N = 20$ for protein sequences). In addition there is a linear gap penalty d, and a given substitution matrix S (e.g. for DNA data the score of an alignment between an a and a g would be the element in row 1, column 2 of this matrix).

We find a highest scoring alignment for two sequences, based upon the highest-scoring alignments of smaller subsequences of x and y.

Denote the segment of the first i elements of x by $x_{1,i}$, and the segment of the first j elements of y by $y_{1,j}$ i.e.

$$x_{1,i} = X_1 X_2 \dots X_i \quad y_{1,j} = Y_1 Y_2 \dots Y_j$$

Label a highest scoring alignment of $x_{1,i}$ and $y_{1,j}$ by $B(i,j)$ for $i = 0, 1, \dots, m$ and $j = 0, 1, \dots, n$. When $j = 0$ (or $i = 0$) this is just the alignment of a sequence of elements against the appropriate number of gaps (each scoring $-d$), so that

$$B(i,0) = -id, \quad B(0,j) = -jd$$

We also set $B(0,0) = 0$.

if we can find all of the $B(i,j)$ we have an $(m+1) \times (n+1)$ matrix, where the entry in the last row and column, $B(m,n)$ is a highest-scoring alignment of the two full sequences.

From above we already know $B(i,0)$ and $B(0,j)$ for all i and j, and we can thus proceed recursively to find $B(i,j)$ in terms of the three elements directly "behind" it,
$B(i-1,j-1), B(i-1,j)$ and $B(i,j-1)$.

Firstly note that a highest scoring alignment can end in three ways. Either
(i) X_i is paired with Y_j,
(ii) X_i is paired with a gap $-$, or
(iii) a gap $-$ is paired with Y_j.

In (i) $B(i,j)$ equals the score of the best alignment between $x_{1,i-1}$ and $y_{1,j-1}$ plus an extra term for the match between X_i and Y_j which we label $s(i,j)$, the element from the substitution matrix which corresponds to the elements X_i and Y_j (thus for DNA if $X_i = Y_j = a$ $s(i,j)$ is the element in row 1 and column 1 of the matrix). Thus we would have
$B(i,j) = B(i-1,j-1) + s(i,j)$.
In (ii) $B(i,j)$ equals the score of the best alignment between $x_{1,i-1}$ and $y_{1,j}$ plus an extra term $-d$ for the alignment of X_i with a gap, so that $B(i,j) = B(i,j-1) - d$.
In (iii) similar reasoning gives
$B(i,j) = B(i-1,j) - d$.

The best alignment is of course the best of these three possibilities, and so

$$B(i,j) = max\{B(i-1,j-1) + s(i,j), B(i-1,j) - d, B(i,j-1) - d\}$$

Using this method we obtain the value of $B(m,n)$. Thus we can find the best possible score with certainty.

The problem is that the running time for this algorithm is of the order mn, so that for very large sequences the time involved can be unrealistically large.

We shall conclude this section (and the course) with a simple example of how the process works.

EXAMPLE

Example: Let $x = gaatct$, $y = catt$ so that $m = 6$ and $n = 4$. We use the simple scoring system of +1 for a match, -1 for a mismatch (the matrix S is 4×4 with every element on the leading diagonal 1, and all others -1) and $d = -2$.

the application of the Needleman Wunsch algorithm is shown in the following figure. Arrows show where each element comes from (in each case there are three possibilities).

FIGURE 20.1

Following the bold arrows gives a highest-scoring alignment, shown below.

g a a t c t
c − a t − t

Note that sometimes more than one arrow comes from a cell, indicating that two possibilities give the same score. By following alternative paths, other highest-scoring alignments can be found, e.g.

g a a t c t
− c a t − t