6 Markov Chains and Hidden Markov Models

(This chapter[1] is primarily based on Durbin et al., chapter 3, [DEKM98] and the overview article by Rabiner [Rab89] on HMMs.)

Why probabilistic models? In problems occurring in bioinformatics, it is often difficult to model objects without imprecisions. A probabilistic model describes a class of objects, each of which occurs with a certain probability.

As an example, consider the following problem: In genomic DNA, the nucleotide pair C-G (C followed by G on the same strand), denoted CpG, is less frequent than could be expected from the frequencies of C and G. The reason is a mechanism called methylation with which CpG pairs are turned into CpT's. However, in certain short areas, for example around promoters or the beginnings of genes, this mechanism is suppressed: CpG occurs more often than in general. These areas are called *CpG islands* and are of particular interest. They are typically a few hundred to a few thousand nucleotides long.

> CpG islands:
> Identify CpG islands. We look at two variants:
>
> 1. Given a short stretch of DNA, is it a CpG island?
> 2. Given a long stretch of DNA, does it contain CpG islands? If so, where are they?

We will solve the first problem using Markov Chains, and the second, using Hidden Markov Models.

6.1 Informal introduction to Markov Chains and HMMs

Let's assume you come from a warm part of the world with a steady climate, say from California, and you ended up here in Germany with its permanently changing weather. You realize that the weather influences your mood, so you try to find out how you can predict it. You also notice after a while that the weather is not actually completely random, but today's weather depends to a certain extent on yesterday's weather conditions. For simplicity, you label each day as having one of the following three weather conditions: sunny, rain/cloudy, snow. After some sampling, you find certain dependencies: If it snowed one day, it will snow the next day with probability 0.2, it will rain with probability 0.5, and it will be sunny with probability

[1] Chapter 6 of Lecture notes by Zsuzsanna Lipták, zsuzsa@cebitec.uni-bielefeld.de, for the course "Selected Topics in Algorithmic Bioinformatics," Winter 2008/09, Bielefeld University - preliminary version of January 27, 2009

0.3. All your findings are summarized in the following table $A = (a_{ij})_{i,j=1,2,3}$, where the entry a_{ij} stands for: The weather will be of type j with probability a_{ij} given that the weather was of type i on the previous day:

	snow	rain	sun
snow	0.2	0.5	0.3
rain	0.1	0.8	0.1
sun	0.1	0.7	0.2

We observe immediately that the rows of matrix A add up to 1: Intuitively, something must happen the next day. The example is visualized in Figure 6.1.

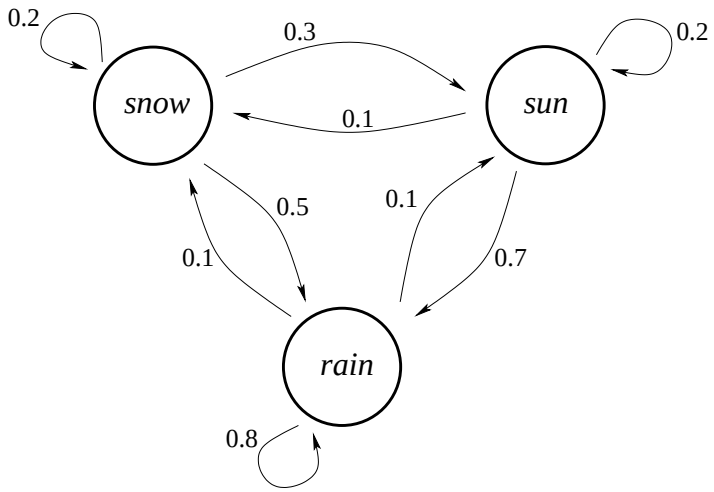

Figure 6.1: Graphical representation of the example Markov Chain.

Say it is Wednesday, it is snowing, and you want to go skiing on Friday for the weekend, and want it to rain on Monday (so as not to feel bad about working again). Therefore, you are interested in the probability of the following sequence: snow (today)-snow-sun-sun-sun-rain. The probability can be easily computed as follows:

$$\begin{aligned} &P(\text{snow today})P(\text{snow}|\text{snow})P(\text{sun}|\text{snow})P(\text{sun}|\text{sun})P(\text{sun}|\text{sun})P(\text{rain}|\text{sun}) \\ &\qquad = 1 \cdot 0.2 \cdot 0.3 \cdot 0.2 \cdot 0.2 \cdot 0.7 = 0.00168. \end{aligned}$$

Up to here, we have a *Markov Chain*, which will be defined precisely in Section 6.3. The states are the weather types; the state at a given time depends only on the previous state; and the observation is the sequence of states.

Now we want to modify the problem. Instead of observing the weather itself, we will only be able to observe something that depends on the states: We now introduce a Hidden Markov Model (HMM).

So, to continue the example, assume that your mood depends to a certain extent upon the weather, namely in the following way. When it is sunny, you are in a good mood 90% of the time. When it is rainy, you are in a bad mood 80% of the time. Moreover, during your first winter in Germany, you also come to love snow. So when it snows, you are in a good mood 70% of the time.[2] This is summarized in the following table $E = (e_i(b))_{i=1,2,3,b\in\{G,B\}}$. Observe that here, the columns add up to 1.

	snow	rain	sun
G	0.7	0.2	0.9
B	0.3	0.8	0.1

Imagine that you talk to your mum on the phone every day. She notices that on some days, you are in a good mood, on others, you are in a bad mood. For no apparent reason. Of course, she lives in California, so it takes her a while to realize that your moods are connected to some extent to the changes in the weather. However, she has no Internet connection, so she does not have any information about the weather itself. Instead, she can only observe your moods. What she observes during one week is, for example: G-G-B-B-B-G-B. She engaged your flatmate some time ago to make notes, so she already has a fair idea of the parameters in the above tables (i.e. the a_{ij}'s and $e_i(b)$'s). We visualize the new situation in Figure 6.2.

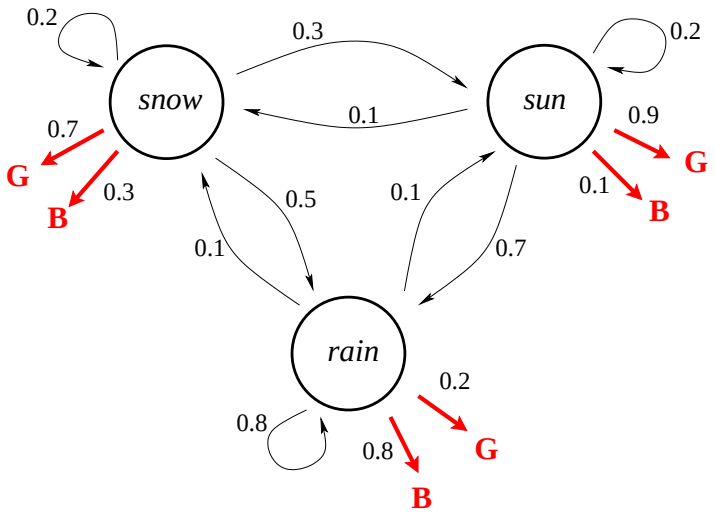

Figure 6.2: Graphical representation of the example Hidden Markov Model.

This is now a Hidden Markov Model. The underlying states (the weather types) still follow a simple Markov Chain, but we can no longer observe them directly. Instead, the observation now consists of a sequence of symbols (the mood types), which are emitted by the states following a certain probability distribution. The formal definition will be given in Section 6.5.

[2] These notes were written during the great snowy period of January 2009.

To complete the example: The questions your mum may ask include:

1. What was the weather like most probably during this week?
2. How likely is it that this sequence will occur again?
3. What are the parameters of the underlying model, i.e. are the parameters A and E correct, or could other parameters better explain the given observations?

We will see an algorithm for answering the first question in Section 6.6, and one for the second question in Section 6.7. We will not answer the last question here but will only sketch the problem.

6.2 A little background on probability

A *finite probability space* is given by

- a finite set Ω (the *sample space* or *ground space*), and
- a function $P : \Omega \to [0,1]$ s.t. $\sum_{\omega\in\Omega} P(\omega) = 1$.

Subsets of Ω are called *events*. Events with cardinality 1 are called *elementary events*. The probability of an event A is defined as $P(A) = \sum_{\omega\in A} P(\omega)$. The complementary event of A is defined as $A^C = \Omega \setminus A$. For two events A, B, the event $A \cap B$ is called the *joint probability* of A and B (often denoted (A, B) instead of $(A \cap B)$). A *partition* of the sample space is a family $A_1, \ldots, A_n \subseteq \Omega$ such that $\cup_i A_i = \Omega$ and $A_i \cap A_j = \emptyset$ for $i \neq j$. Let A, B be events. Then

1. $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.
2. $P(A^C) = 1 - P(A)$.
3. $P(A \cap B) \geq P(A) - P(B^C)$.
4. If $B_1, \ldots, B_n$ is a partition of Ω, then $P(A) = \sum_{i=1}^{n} P(A \cap B_i)$.

Example 5. A fair die is thrown. Then $\Omega = \{1,2,3,4,5,6\}$, and $P(x) = \frac{1}{6}$ for all $x \in \Omega$. Let $A = \{2,4\}$, $B = \{3,4,5\}$. Then $P(A) = \frac{2}{6}, P(B) = \frac{3}{6}, P(A \cup B) = P(A) + P(B) - P(A \cap B) = \frac{2}{6} + \frac{3}{6} - \frac{1}{6} = \frac{4}{6}$. Let $B_1 = \{\text{result is even}\}$ and $B_2 = \{\text{result is odd}\}$. Then B_1, B_2 is a partition of Ω.

6.2.1 Conditional probabilities and Bayes' Theorem

For two events A, B with $P(B) \neq 0$, the *conditional probability of* A *given* B is defined as

$$P(A \mid B) \quad = \quad \frac{P(A \cap B)}{P(B)} \tag{6.1}$$

We call A and B *independent* if $P(A \mid B) = P(A)$. (This is the case if and only if $P(B \mid A) = P(B)$). Another equivalent statement is that $P(A \cap B) = P(A)P(B)$. As an exercise, you can show that these three statements are equivalent.)

Example 6. Let (Ω, P) as before, and $A = \{2\}$, $B = \{\text{result is even}\}$. Then $P(A \mid B) = \frac{1}{3}, P(B \mid A) = 1$.

The *theorem of total probability* states that if $B_1, \ldots, B_n$ is a partition of Ω, then

$$P(A) = \sum_i P(A \mid B_i)P(B_i). \tag{6.2}$$

The following is referred to as *Bayes' theorem* or *Bayes' formula*: Given two events A and B s.t. $P(B) \neq 0$,

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}. \tag{6.3}$$

Bayes' formula is a simple consequence of the definition of conditional probabilities, but it has profound consequences. In particular, it gives us a method for deciding between alternative models for the same experimental data, see the next section.

6.2.2 How to use Bayes' theorem for model comparison

Here we show how to decide between alternative models. The question is: Given (experimental) data, which model (what hypothesis) explains the data better?

Consider the following example (taken from Durbin [DEKM98], p.6.): In a casino, there are two types of dice, fair and loaded. A fair die produces $1, 2, 3, 4, 5, 6$ each with probability $\frac{1}{6}$. A loaded die shows a 6 with probability $\frac{1}{2}$, and all other numbers with probability $\frac{1}{10}$ each. 99% of dice are fair, 1% loaded. We pick up a die at random and roll it 3 times. We observe the sequence $6, 6, 6$. Is the current die fair or loaded?

So we have $P(F) = 0.99$, $P(L) = 0.01$. Further, $P(x \mid F) = \frac{1}{6}$ for $x \in \{1, 2, 3, 4, 5, 6\}$, while $P(6 \mid L) = \frac{1}{2}$ and $P(x \mid L) = \frac{1}{10}$ for $x \in \{1, 2, 3, 4, 5\}$. We can also compute the joint probabilities $P(6, F) = \frac{1}{6} \cdot \frac{99}{100} = \frac{99}{600}$ and $P(6, L) = \frac{1}{2} \cdot \frac{1}{100} = \frac{3}{600}$.

In order to judge whether our die is fair or loaded, we want to work out the *posterior probability* $P(\text{model} \mid \text{data})$, in this case of the hypothesis that the die is loaded, $P(L \mid 3 \text{ sixes})$. By Bayes' theorem,

$$P(L \mid 3 \text{ sixes}) = \frac{P(3 \text{ sixes} \mid L)P(L)}{P(3 \text{ sixes})}. \tag{6.4}$$

$P(3 \text{ sixes} \mid L)$ is called the *likelihood* of the hypothesis. By independence, we have $P(3 \text{ sixes} \mid L) = P(6 \mid L)^3 = \frac{1}{8}$. Moreover, by the theorem of total probability, $P(3 \text{ sixes}) = P(3 \text{ sixes} \mid L)P(L) + P(3 \text{ sixes} \mid F)P(F) = \frac{1}{8} \cdot \frac{1}{100} + (\frac{1}{6})^3 \cdot \frac{99}{100} = 0.00125 + 0.00458333\ldots \approx 0.0058$. So altogether we have

$$P(L \mid 3 \text{ sixes}) = \frac{P(3 \text{ sixes} \mid L)P(L)}{P(3 \text{ sixes})} = \frac{0.00125}{0.0058} \approx 0.214. \tag{6.5}$$

So in spite of the unlikely outcome of 3 sixes, we are still more likely to have a fair die than a loaded die.

When we have no prior information about the probability of the different models, then it suffices to look at the likelihood $P(\text{data} \mid \text{model})$. More precisely: Let the data D be given, and let $M_1, \ldots, M_k$ be different models. We want to choose the one which explains the data best, so we are interested in which model maximizes the posterior probability $P(M_i \mid D)$. If we do not know or do not want to assume anything about $P(M_i)$ for the different models M_i, then we should assume uniform distribution, i.e., $P(M_i) = P(M_j)$ for all i, j. In this case (but only in this case!) we have: $P(M_i \mid D) > P(M_j \mid D)$ if and only if $P(D \mid M_i) > P(D \mid M_j)$, because $P(M_i \mid D) = \frac{P(D|M_i)P(M_i)}{P(D)} = \frac{P(D|M_i)P(M_j)}{P(D)} > \frac{P(D|M_j)P(M_j)}{P(D)} = P(M_j \mid D)$. (This proves the implication $P(D \mid M_i) > P(D \mid M_j) \Rightarrow P(M_i \mid D) > P(M_j \mid D)$, which is what we need here.)

Thus, in cases where there is no prior information about the distribution of the models, then one looks at the likelihoods $P(\text{data} \mid \text{model})$, see e.g. Section 6.4.

6.3 Formal definition of Markov Chains

A (first order, discrete, homogeneous) Markov Chain (M.C.) is a stochastic process consisting of

- a finite set of states Q, where $|Q| = N$; we will refer to states as $1, 2, \ldots N$. We denote the state at time i as q_i.
- a transition probability matrix $A = (a_{k\ell})_{k,\ell \in Q}$ of size $N \times N$, where $a_{k\ell}$ is the probability of being in state ℓ, given that the previous state was k:

  $$a_{k\ell} = P(q_i = \ell \mid q_{i-1} = k).$$

  So $0 \leq a_{k\ell} \leq 1$ for all $k, \ell \in Q$, and $\sum_\ell a_{k\ell} = 1$ for all k (the rows add up to 1). Note that the $a_{k\ell}$'s are independent of time i (thus, the M.C. is homogeneous).
- an initial probability vector $\pi = (\pi_1, \ldots, \pi_N)$, where

  $$\pi_k = P(q_1 = k), \text{ the probability of starting in state } k.$$

  Thus, $0 \leq \pi_k \leq 1$ for all k, and $\sum_k \pi_k = 1$.

The Markov chain can be written as a triple $\mathcal{M} = (Q, A, \pi)$. It is common to visualize a Markov chain as a directed graph with loops, see Fig. 6.3.

Since $\mathcal{M}$ is a first order Markov chain, the probability of being in state k depends only on the previous state, in other words

$$P(q_i = k_i \mid q_1 = k_1, \ldots, q_{i-1} = k_{i-1}) = P(q_i = k_i \mid q_{i-1} = k_{i-1}). \tag{6.6}$$

Equation (6.6) is often referred to as the *Markov property*, and is the fundamental property of (first-order) Markov Chains. Using the Markov property, it is easy to compute the probability of a path $q = q_1 \ldots q_L$:

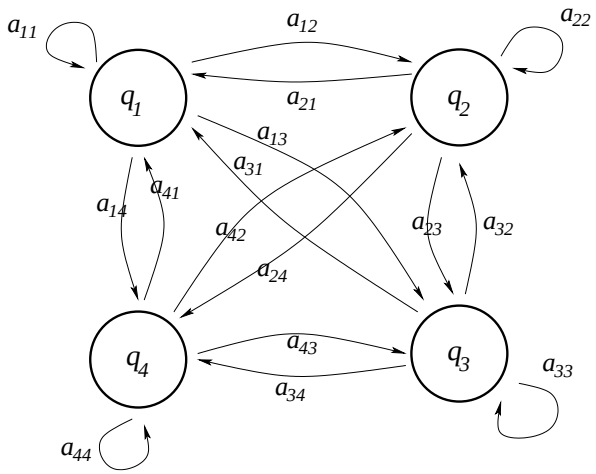

Figure 6.3: Graphical representation of a Markov chain with 4 states.

$$P(q_1, q_2, \ldots, q_L) = \pi_{q_1} \prod_{i=2}^{L} a_{q_i q_{i-1}}. \tag{6.7}$$

Note that we are using a slightly sloppy notation: More formally we should write $P(q_1 = k_1, q_2 = k_2 \ldots, q_L = k_L)$ instead of $P(q_1, q_2, \ldots, q_L)$, but whenever the other variable is not necessary, we drop it (in this case k_i, the value of the state at time i). So $P(q_i)$ stands for: the probability of the state at time i being what it is, namely the value of q_i. (More precisely, the states can be modelled as random variables.)

Example 7. Consider the example from Section 6.1. Given the initial probability vector $\pi = (0.1, 0.7, 0.2)$, the probability of the sequence snow-sun-sun is $P(\text{snow-sun-sun}) = 0.1 \cdot 0.3 \cdot 0.2 = 0.006$.

The following are important properties of Markov chains:

1. A^n gives the transition probabilities after n steps, i.e., if $a_{k\ell}^{(n)}$ is the (k, ℓ)'th entry of A^n, then $a_{k\ell}^{(n)} = P(q_{t+n} = \ell \mid q_t = k)$, the probability of being in state ℓ at time $t + n$ if we were in state k at time t (for all $t \geq 0$).

2. The distribution after n steps, starting from the initial distribution π, can be computed as πA^n.

6.3.1 Moving from probabilities to log of probabilities

When computing probabilites, we often have to multiply many very small numbers (e.g. when computing the probability of a particular path q using Equation (6.7)). This quickly leads to computational problems in form of rounding errors. The problem is usually solved by moving from probabilities to the log of probabilities, so Equation (6.7) becomes

$$\log P(q_1, q_2, \ldots, q_L) = \log \pi_{q_1} + \sum_{i=2}^{L} \log a_{q_i q_{i-1}}. \tag{6.8}$$

Hereby, any log function can be used, but $\log_2$, $\ln = \log_e$ and $\log_{10}$ are the most common. We recall the following fundamental properties of the logarithm:

Lemma 6.1 (Some properties of the log-function)**.** *Let* $\log(x) = \log_b(x)$ *denote the logarithm to some base* $b > 1$*. Let* $x, y > 0$*. We have*

1. $\log(xy) = \log x + \log y$.
2. $\log(\frac{x}{y}) = \log x - \log y$.
3. log *is strictly monotonically increasing, i.e.,* $x > y$ *if and only if* $\log x > \log y$.
4. *In particular, the mapping* $\log : (0, +\infty) \to (-\infty, +\infty)$ *is a bijection.*

The first two properties will simplify computation, and the last two properties ensure that we can freely move back and forth between probabilities and log-probabilities, including when we have to maximize probabilities (see e.g. Sections 6.4 and 6.6).

6.4 Model comparison with Markov chains

Let's return to our Problem 1 on CpG islands: Given a short DNA sequence x, is it a CpG island? We will construct two Markov chains, one modelling CpG islands (the $+$ model), the other non-CpG-islands (the $-$ model), and compare their likelihoods $P(x \mid + \text{model})$ and $P(x \mid - \text{model})$. (Because we have no information about the frequency of CpG islands, we assume that both models are equally likely.) We will use a *likelihood ratio test*.

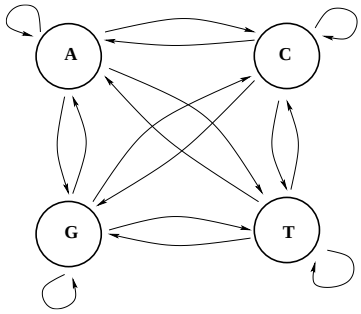

Figure 6.4: Markov chain for describing nucleotide sequences.

Each Markov chain has four states, A,C,T, and G, see Figure 6.4. For estimating the transition probabilities, we use maximum likelihood (ML) estimators: Using a database of DNA sequences, where the CpG islands are known, we denote by c_{ij}^{+} the absolute frequency of nucleotide j following nucleotide i *within* a CpG island. Then

+	A	C	G	T
A	0.180	0.274	0.426	0.120
C	0.171	0.368	0.274	0.188
G	0.161	0.339	0.375	0.125
T	0.079	0.355	0.384	0.182

−	A	C	G	T
A	0.300	0.205	0.285	0.210
C	0.322	0.298	0.078	0.302
G	0.248	0.246	0.298	0.208
T	0.177	0.239	0.292	0.292

Figure 6.5: Transition probabilities for CpG islands (+) and non CpG islands (−), taken from [DEKM98].

we set a_{ij}^+ as the fraction of i followed by j over all pairs ik (k any nucleotide). We compute a_{ij}^- analogously using c_{ij}^-, the frequency of nucleotide j following nucleotide i *outside* a CpG island:

$$a_{ij}^+ = \frac{c_{ij}^+}{\sum_{k\in Q} c_{ik}^+} \quad \text{and} \quad a_{ij}^- = \frac{c_{ij}^-}{\sum_{k\in Q} c_{ik}^-}. \tag{6.9}$$

The resulting transition tables for the + and the − models are given in Figure 6.5. As expected, a_{CG}^+ is much higher than a_{CG}^-. To test whether our sequence x is a CpG island, we compute $P(x \mid + \text{model})$ and $P(x \mid - \text{model})$ and compare which is higher. Or, reducing the two computations to one, we can compute the *log-odds-ratio* $S(x)$ of $x = x_1 \dots x_L$ w.r.t. the two models as

$$\begin{aligned} S(x) &= \log \frac{P(x \mid + \text{model})}{P(x \mid - \text{model})} = \log \frac{\prod_{i=1}^L a_{x_{i-1}x_i}^+}{\prod_{i=1}^L a_{x_{i-1}x_i}^-} = \log \prod_{i=1}^L \frac{a_{x_{i-1}x_i}^+}{a_{x_{i-1}x_i}^-} \\ &= \sum_{i=1}^L \log \frac{a_{x_{i-1}x_i}^+}{a_{x_{i-1}x_i}^-} = \sum_{i=1}^L (\log a_{x_{i-1}x_i}^+ - \log a_{x_{i-1}x_i}^-), \end{aligned} \tag{6.10}$$

where, for convenience of notation, we define $x_0 = 0$ and $a_{0i}^+ = \pi_i^+$, and analogously for the − model. Now we have

$$P(x \mid + \text{model}) > P(x \mid - \text{model}) \Leftrightarrow \frac{P(x \mid + \text{model})}{P(x \mid - \text{model})} > 1 \Leftrightarrow \log \frac{P(x \mid + \text{model})}{P(x \mid - \text{model})} > 0.$$

Therefore, if $S(x) > 0$, then x is more likely to be a CpG island than a non CpG island.

6.5 Hidden Markov Models

Now we turn to Problem 2 on CpG islands: Given a long DNA sequence x (of length L), where are the CpG islands contained in it, if any? A naive solution would be: Use the two Markov chains defined above and test, for each short substring of x (i.e., of length ℓ for fixed values of ℓ), whether it is a CpG island. Problem: What values of ℓ should we choose? Instead, we will use a Hidden Markov Model for solving the problem.

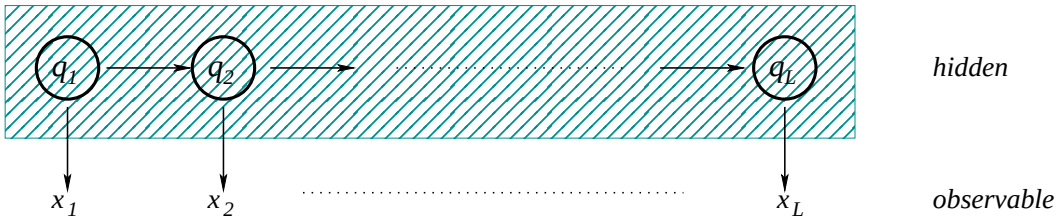

Figure 6.6: Graphical representation of a Hidden Markov Model.

A Hidden Markov Model (HMM) is a triple $\mathcal{H} = (\Sigma, Q, \Theta)$ where

- Σ is a (finite) alphabet of symbols. Denote the symbol emitted at time i by x_i.
- Q is a finite set of states, $|Q| = N$.
- Θ are the parameters:
  - state transition matrix $A = (a_{k\ell})_{k,\ell \in Q}$, where $a_{k\ell} = P(q_i = \ell \mid q_{i-1} = k), i \geq 1$.
  - emission probabilities $e_k(b)$, $k \in Q, b \in \Sigma$, where $e_k(b) = P(x_i = b \mid q_i = k) = P(b \mid k), i \geq 1$.
  - an initial state distribution vector $\pi = (\pi_1, \ldots, \pi_N)$.

Note that the $a_{k\ell}$ and the vector π are probability distributions, thus for all k, ℓ, $0 \leq a_{k\ell}, \pi_k \leq 1$. Moreover, $\sum_{k=1}^{N} \pi_k = 1$, and $\sum_{\ell=1}^{N} a_{k\ell} = 1$ for all k.

Example 8. In the introductory example (Section 6.1), $\Sigma = \{G, B\}$ and $Q = \{\text{snow,rain,sun}\}$.

A *path* is a sequence of states $q = q_1 q_2 \ldots q_L$, which follows a simple Markov model $\mathcal{M} = (Q, A, \pi)$. The path q emits a sequence of symbols $x = x_1 x_2 \ldots x_L \in \Sigma^L$, which is referred to as the *observation*. In a HMM, there is no one-to-one correspondence between an observation x and the underlying state path q. Hence the term *hidden*.

If we know both the observation sequence $x = x_1 \ldots x_L$ and the state path $q = q_1 \ldots q_L$, then it is easy to compute the joint probability $P(x, q)$:

$$P(x, q) = \pi_{q_1} e_{q_1}(x_1) \prod_{i=2}^{L} a_{q_{i-1} q_i} e_{q_i}(x_i). \tag{6.11}$$

However, usually we do not know q, but only know the observation x. Given an observation x, we want to answer the following questions:

1. What is a most likely path that could have generated x?
2. What is the probability of x?
3. What is the probability of being in state k at time i, given the observation x, i.e., what is $P(q_i = k \mid x)$, for fixed k and i? (Motivation for this last question will be given later.)

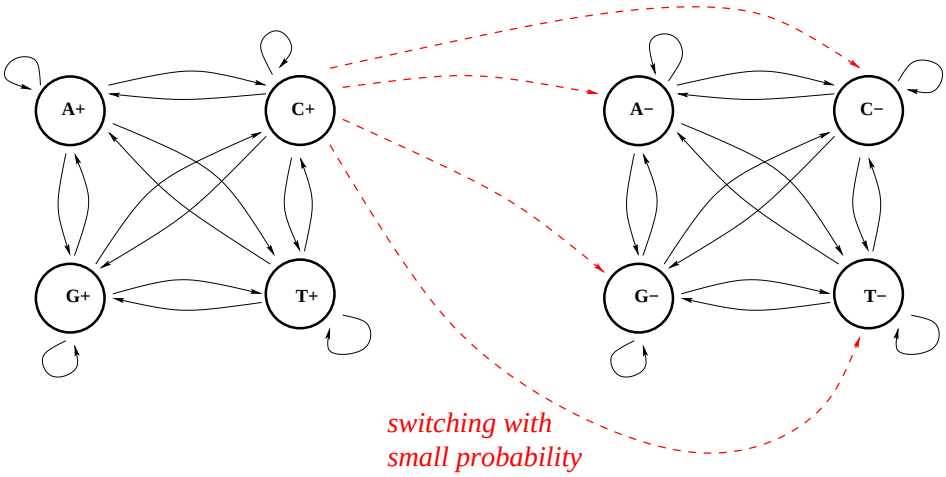

Figure 6.7: HMM for the CpG island example: Of the new transitions, only the transitions from C^+ to the $-$ states (dashed) are included in the drawing.

	A^+	C^+	G^+	T^+	A^-	C^-	G^-	T^-
A^+	0.180p	0.274p	0.426p	0.120p	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$
C^-	0.171p	0.368p	0.274p	0.188p	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$
G^-	0.161p	0.339p	0.375p	0.125p	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$
T^-	0.079p	0.355p	0.384p	0.182p	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$
A^-	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	0.300q	0.205q	0.285q	0.210q
C^-	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	0.322q	0.298q	0.078q	0.302q
G^-	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	0.248q	0.246q	0.298q	0.208q
T^-	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	0.177q	0.239q	0.292q	0.292q

Figure 6.8: Transition probabilities the HMM for identifying CpG islands, derived from the transition tables in Fig. 6.5 (taken from [Sha01]).

An answer to 2. will supply us with an answer to our Problem no. 2 on CpG-islands. We model the problem as an HMM by combining the two previous Markov chains.

The transition matrix is given in Fig. 6.8, assuming that the probability of staying within a CpG island is p, and that of staying outside is q. Moreover, we assume that when switching between the two types of DNA, each nucleotide is equally likely (i.e., $P(i^- \mid j^+) = \frac{1-p}{4}$ and $P(i^+ \mid j^-) = \frac{1-q}{4}$ for any $i, j \in \{A, C, T, G\}$). Each state emits only one symbol, namely $e_{i^+}(i) = e_{i^-}(i) = 1$ and $e_{i^+}(j) = e_{i^-}(j) = 0$ for $j \neq i$, where $i, j \in \{A, C, G, T\}$. This makes the HMM slightly degenerate[3]. Thus the paths $G^+G^+C^+G^+$, $G^-G^-C^-G^-$ and $G^+G^-C^-G^+$ all emit the sequence of symbols GGCG, but are not all equally likely.

[3]Modelling the problem with only two hidden states + and - is not possible because then we would lose the dependence of the symbols on the previously emitted symbol.

6.6 Computing the most probable path

Given observation $x = x_1 x_2 \dots x_L$, we want to find a most probable state path $q^* = q_1^* \dots q_L^*$ that could have emitted x. In other words, we are looking for

$$q^* = \operatorname*{argmax}_q P(x, q). \tag{6.12}$$

Note that q^* need not be unique, i.e., there may be more than one path q maximizing $P(x, q)$. In that case, q^* can be chosen arbitrarily as any of these q's. For simplicity of exposition, we will assume from now on that q^* is unique.

We can find q^* by enumerating all possible paths q of length L, computing the joint probability $P(x, q)$ of q and x, and choosing the one(s) with highest probability. However, this method is inefficient since the number of paths grows exponentially with L. Instead, q^* can be found recursively. We will use dynamic programming (DP) for finding q^*. Fix state k and time i, and consider

$$\begin{aligned} V_k(i) \quad = \quad & \text{highest probability of any path } q = q_1 \dots q_i \\ & \text{emitting } x_1 \dots x_i, \text{ where } q_i = k, \end{aligned}$$

the probability of a most probable path ending in k that emitted x up to and including x_i. (Note that the $V_k(i)$'s are defined dependent on the observation x, so you get different $V_k(i)$'s for different x.) If we can compute the $V_k(i)$'s, then we are done, because

$$P(x, q^*) = \max\{V_k(L) \mid k \in Q\}, \tag{6.13}$$

and q^* can be found by backtracing in the DP table. But the $V_k(i)$ can be computed from previous values, namely for any $k \in Q$ and $i \geq 2$,

$$V_k(i) = \underbrace{e_k(x_i)}_{\substack{\text{prob. of } k \\ \text{emitting } x_i}} \cdot \max_\ell \left(\underbrace{V_\ell(i-1)a_{\ell k}}_{\substack{\text{prob. of best path which ends in } \ell \\ \text{and of transition from } \ell \text{ to } k}} \right). \tag{6.14}$$

Now all we need are the initial values $V_k(1) = e_k(x_1)\pi_k$, and we are ready for the algorithm.

Viterbi's algorithm

Initialization ($i = 1$):	$V_k(1) = e_k(x_1)\pi_k$	for all $k \in Q$
Recursion ($i = 2, \dots, L$):	$V_k(i) = e_k(x_i) \max_\ell (V_\ell(i-1)a_{\ell k})$	for all $k \in Q$
	$\text{ptr}_i(k) = \operatorname{argmax}_\ell (V_\ell(i-1)a_{\ell k})$	// remember path
Termination:	$P(x, q^*) = \max_k V_k(L)$	
	$q_L^* = \operatorname{argmax}_k (V_k(L))$	
Traceback ($i = L, \dots, 2$):	$q_{i-1}^* = \text{ptr}_i(q_i^*)$.	// recover path q^*

Complexity

We need to store the DP table which has size $N \cdot L$, where $N = |Q|$ is the number of states, and L the length of the observation, so storage space is $O(NL)$. For each entry $V_k(i)$, we maximize over N values $a_{\ell k} \cdot V_\ell(i-1)$, so runtime of the algorithm is $O(N^2 L)$ for computing $P(x, q^*)$, and $O(NL)$ for the traceback step (producing q^*).

(Compare to $O(N^L L)$ time for computing $P(q, x)$ for *every* possible path q, the naive solution mentioned at the beginning of this section.)

6.7 Posterior decoding

Recall that we want to have information about the hidden state path, given our observation sequence x. Defining q^*, the most probable path, is only one possible alternative. If many different paths have almost equal probability of emitting x, it may not be very informative to have q^*. Instead, we may be interested in the most probable state *at a particular time* i, given observation x. We will look for the path $\hat{q} = (\hat{q}_1, \ldots, \hat{q}_L)$ s.t. for each i, $\hat{q}_i$ is the most likely state, given observation x.

For each $i = 1, \ldots, L$, let us find the state k which maximizes $P(q_i = k \mid x)$, i.e., the most probable state at time i, given x:

$$\hat{q}_i = \operatorname*{argmax}_k P(q_i = k \mid x). \tag{6.15}$$

Note that $\hat{q}$ and q^* need not be identical; in fact, $\hat{q}$ may not even be a legal path (if not all $a_{k\ell}$ are positive, then there are impossible transitions).

How do we compute $P(q_i = k \mid x)$? Recall that for two events A, B, where $P(B) \neq 0$, we have 1. $P(A \mid B) = \frac{P(A \cap B)}{P(B)}$, and 2. $P(A \cap B) = P(A)P(B \mid A)$.

$$P(q_i = k \mid x) \stackrel{1.}{=} \frac{P(q_i = k, x)}{P(x)}. \tag{6.16}$$

We will return in a moment to computing $P(x)$. The enumerator $P(q_i = k, x)$ can be computed as

$$\begin{aligned} P(q_i = k, x) &= P(\overbrace{q_i = k, x_1, x_2, \ldots, x_i}^{A}, \overbrace{x_{i+1}, \ldots, x_L}^{B}) \\ &\stackrel{2.}{=} P(x_1, \ldots, x_i, q_i = k) P(x_{i+1}, \ldots, x_L \mid x_1, \ldots, x_i, q_i = k) \\ &= \underbrace{P(x_1, \ldots, x_i, q_i = k)}_{\text{forward variable } F_k(i)} \underbrace{P(x_{i+1}, \ldots, x_L \mid q_i = k)}_{\text{backward variable } B_k(i)}, \end{aligned} \tag{6.17}$$

where the last equality holds because the emission at time t depends only on the state at time t (and not on previous emissions), and the state at time t depends only on the state at time $t-1$. Altogether we have

$$P(q_i = k \mid x) = \frac{F_k(i) B_k(i)}{P(x)}. \tag{6.18}$$

6.7.1 The forward variable. Computing $P(x)$.

Computing the forward variable is easy, in fact it is almost identical to computing the $V_k(i)$'s. The crucial observation is

$$\begin{aligned} F_k(i) &= P(x_1,\ldots,x_i,q_i=k) = \underbrace{e_k(x_i)}_{\substack{\text{prob. of } k\\ \text{emitting } x_i}} \cdot \sum_{\ell\in Q} \underbrace{P(x_1,\ldots,x_{i-1},q_{i-1}=\ell)\cdot a_{\ell k}}_{\substack{\text{prob. of a path emitting } x_1,\ldots,x_{i-1},\\ \text{ending in } \ell, \text{ and of transition from } \ell \text{ to } k}} \\ &= e_k(x_i)\sum_{\ell\in Q} F_\ell(i-1)a_{\ell k}, \end{aligned} \tag{6.19}$$

so the max in Equation (6.14), the recursion for $V_k(i)$, is replaced by a sum. The initial condition is $F_k(1) = e_k(x_1)\pi_k$ for all $k \in Q$.

Moreover, we can now easily compute $P(x)$, the probability of the sequence x occuring in the given HMM (see Question 2 from Section 6.5):

$$P(x) = \sum_{k\in Q} F_k(L). \tag{6.20}$$

The full algorithm is given below.

Forward algorithm

Initialization ($i = 1$):	$F_k(1) = e_k(x_1)\pi_k$	for all $k \in Q$
Recursion ($i = 2,\ldots,L$):	$F_k(i) = e_k(x_i)\sum_\ell F_\ell(i-1)a_{\ell k}$	for all $k \in Q$
Termination:	$P(x) = \sum_k F_k(L)$	

Again, runtime is $O(N^2L)$ and space $O(NL)$.

6.7.2 The backward variable

The backward variable is a bit less intuitive than the forward variable. It is called backward variable, because the DP table is filled in from back to front, i.e., starting from the L'th column, and ending at the first. Note that the definition of $B_k(i)$ is a *conditional* probability (as opposed to the forward variable, which is a joint probability): $B_k(i)$ is the probability that the remaining sequence $x_{i+1}\ldots x_L$ will be produced, given that the current state q_i is k. We can compute this value if we already know the later values $B_\ell(i+1)$ for $\ell \in Q$:

$$\begin{aligned} B_k(i) &= P(x_{i+1},\ldots,x_L \mid q_i = k) \\ &= \sum_\ell a_{k\ell}\cdot e_\ell(x_{i+1})\cdot P(x_{i+2}\ldots x_L \mid q_{i+1}=\ell) \\ &= \sum_\ell a_{k\ell}\cdot e_\ell(x_{i+1})\cdot B_\ell(i+1). \end{aligned} \tag{6.21}$$

The extremal conditions are $B_k(L) = 1$ for all $k \in Q$. Again, the backward variables can be used to compute $P(x)$, and we have

$$P(x) = \sum_{k \in Q} B_k(1) \cdot e_k(x_1)\pi_k. \tag{6.22}$$

Here is the full algorithm:

Backward algorithm

Initialization ($i = L$):	$B_k(L) = 1$	for all $k \in Q$
Recursion ($i = L-1, \ldots, 1$):	$B_k(i) = \sum_\ell a_{k\ell} e_\ell(x_{i+1}) B_\ell(i+1)$	for all $k \in Q$
Termination:	$P(x) = \sum_k B_k(1) e_k(x_1)\pi_k$	

Again, runtime is $O(N^2L)$ and space $O(NL)$.

6.8 Other topics

Another big topic is how to estimate the parameters of a given HMM. This is done using so called *training sequences* $x_1, \ldots, x_n$, which are then used in order to find good values for the parameters of the HMM. Two cases can be distinguished, namely when the state paths are known, and when the state paths are not known.

If the state paths are known, ML estimators can be used: We define a_{ij}'s as we did in Section 6.4, by setting a_{ij} the fraction of transitions from state i to j over all transitions from i to any state k. The initial probability vector π and the emission probabilities $e_k(b)$ can be estimated similarly. Often, ML estimators are corrected by adding some base value in order to avoid overfitting.

When the state paths are not known, iterative algorithms such as Viterbi training or the Baum-Welch algorithm are used. The idea is to start with an arbitrary set of parameters and iteratively improve them until reaching some local optimum.

For more, see Durbin et al., chapter 3, [DEKM98] and the overview article by Rabiner [Rab89] on HMMs.