

What Are the Chances?—Hidden Markov Models



Angela B. Shiflet, George W. Shiflet, Mario Cannataro, Pietro Hiram Guzzi, Chiara Zucco, and Dmitriy A. Kaplun

Abstract Hidden Markov Models (HMMs) can be used to solve a variety of problems from facial recognition and language translation to animal movement characterization and gene discovery. With such problems, we have a sequence of observations that we are not certain is correct—we are not sure our observations accurately reveal the corresponding sequence of actual states, which are hidden—but we do know some important probabilities that will help us. In this chapter, we will develop the probability theory and algorithms for two types of problems that HMMs can solve—calculate the probability that a particular sequence of observations occurs and determine the most likely corresponding sequence of hidden states. The chapter will end with a collection of research projects appropriate for undergraduates.

Suggested Prerequisites *Previous programming experience; a basic understanding of probability; knowledge of Markov models, such as in Module 13.4, “Probable Cause: Modeling with Markov Chains,” from [19]; understanding of parallel programming for optional Sect. 9, “Parallel Forward Algorithm,” and optional Sect. 10.3, “Parallel Viterbi Algorithm” and the corresponding projects.*

1 Introduction

We can employ **Hidden Markov Models (HMMs)** to solve a variety of problems from facial recognition and language translation to animal movement characteriza-

A. B. Shiflet (✉) · G. W. Shiflet · D. A. Kaplun
Wofford College, Spartanburg, SC, USA
e-mail: shifletab@wofford.edu; shifletgw@wofford.edu; kaplunda@email.wofford.edu

M. Cannataro · P. H. Guzzi · C. Zucco
University “Magna Græcia” of Catanzaro, Catanzaro, Italy
e-mail: cannataro@unicz.it; hguzzi@unicz.it

tion and gene discovery (see “[Further Reading](#)”). In such a problem, we know a sequence of observations, but we are not certain that our observations are accurate; the actual sequence of states is hidden, or unknown. However, we often know for each state the probability that it is the initial one, the probabilities of transitioning from any one state to every other state, and the probabilities of a state resulting in each type of observation. Hidden Markov Models can attack three types of problems:

- Likelihood problem: Calculate the probability of a particular sequence of observations
- Decoding problem: For a particular sequence of observations, determine the most likely underlying sequence of states
- Training problem: For a sequence of observations and a sequence of hidden states, discover the most likely HMM parameters

In this article, we cover the basics of Hidden Markov Models, algorithms, and applications involving the first two types of problems. We begin with an example of how HMMs are used to advance our understanding of genetics and the workings of the human body.

1.1 Case in Point

Nina was 15 years old and 50 inches tall. Her mother was concerned that her daughter was so short. The Centers for Disease Control and Prevention growth charts indicate a range for normal height for 15-year-old females to be between 60 and 68 inches [2]. Testing done from her physical examination indicated that Nina has growth hormone deficiency. Children diagnosed with this deficiency have inadequate secretion of **growth hormone (GH)**, a hormone produced and stored by **somatotropic cells** of the pituitary gland at the base of the brain. Growth hormone molecules are stored in the pituitary until **growth hormone-releasing hormone (GHRH)** is secreted by the brain. The secreted GHRH binds specifically to **growth hormone-releasing hormone receptors**, found within the cell membranes that form the surface of the somatotropic cells. The receptor is associated with a protein, called a **G-protein**, which transduces the external signal into internal chemical signals (second messengers). The second messengers induce the cell to release GH.

Because Nina’s brother was given the same diagnosis, both patients were referred for genetic evaluation. The results of these tests indicated that both children had a genetic mutation in the gene that codes for the GHRH receptor. This mutation may alter the receptor enough that it no longer binds to the signal as well, and therefore there is less internal response to the signal. Hence, less GH is released into the bloodstream. Children with growth hormone deficiency are generally treated with periodic injections of GH.

The mechanism for the release of growth hormone from the pituitary is very complex. Much is still unknown about G-proteins and their receptors, so scientists

are working to unravel these mysteries. So, how do G-proteins and receptors relate to hidden Markov models? One example is found in experiments conducted at the University of Birmingham, UK [22]. Using sophisticated imaging studies of individual G-proteins and receptors, researchers could track and map the diffusion of particles (proteins) along the membrane. Using HMM, they assumed that particles shift among discrete diffusive states that follow a randomly determined course. They found that G-proteins and receptors move through four discernable, diffusive states, varying from immobile to fast diffusing, and have association with four different diffusion coefficients, which are the possible observations. Their results were consistent with results of the imaging studies. From these experiments and analyses, the scientists concluded that G-proteins and their receptors are sequestered into small membrane compartments. Such restriction of diffusion allows the two particles to bind more easily, although the binding is very short-lived.

From these results we understand more about how these important components in cell signaling work. As we figure out the intricacies of cell signaling, we may discover ways to modify or correct defects in the components.

2 Example Model

Although the studies of human growth factor are quite interesting, there are many other more approachable problems. Moreover, we will consider other genetic applications of HMMs later in this chapter. Thus, we begin our study of the mathematics of HMMs with a simpler hypothetical example involving animal behavior.

With a primary diet of leaves, which are not very nutritional and are hard to digest, red howler monkeys spend most of their time eating and resting. Suppose scientists in a simplified appraisal considered the monkey to be in two primary states, eating (E) or resting/sleeping (R), so that the **set of possible states** is $S = \{E, R\}$. Moreover, suppose the biologists observe that, on average, the monkeys spend 30% of their time eating and 70% sleeping or resting. In this case, where u_0 is the state at time 0 and P indicates probability, the **initial state probability**, π , for E is $\pi(E) = P(u_0 = E) = 0.30$, and the initial state probability for R is $\pi(R) = P(u_0 = R) = 0.70$. (As we will see later, the choice of variable, u , represents the “underlying,” or hidden, state.)

Suppose also the biologists determined that if a monkey is eating at hour k , so that its state is $u_k = E$, then there is a 60% chance that the animal will be eating the next hour ($k + 1$), or $u_{k+1} = E$. Thus, in terms of conditional probability, $P(u_{k+1} = E | u_k = E) = 0.6$ and $P(u_{k+1} = R | u_k = E) = 1 - 0.6 = 0.4$; that is, given $u_k = E$, the probability of $u_{k+1} = E$ is 0.6, and the probability of $u_{k+1} = R$ is 0.4. Suppose the scientists also discovered that $P(u_{k+1} = E | u_k = R) = 0.2$, so that $P(u_{k+1} = R | u_k = R) = 1 - 0.2 = 0.8$. If resting at hour k , the animal has a 20% chance of eating and an 80% chance of resting the next hour. Figure 1 presents a state diagram of the findings, with probabilities of transitioning from one

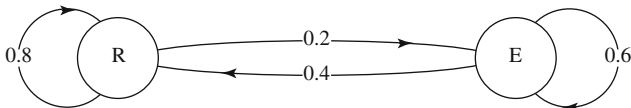


Fig. 1 State diagram for hypothetical study on red howler monkeys

state to another on the arrows. Thus, the following **transition matrix** summarizes their findings:

$$T = \begin{matrix} & u_k/u_{k+1} & \begin{matrix} E & R \end{matrix} \\ \begin{matrix} E \\ R \end{matrix} & \begin{bmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{bmatrix} \end{matrix}.$$

We can also denote these **transition probabilities**, t , as $t(\text{previous state}, \text{next state})$, so $t(E, E) = 0.6$, $t(E, R) = 0.4$, $t(R, E) = 0.2$, and $t(R, R) = 0.8$. The transition probabilities (or, comparably, matrix T) form a **Markov model** with each state, u_{k+1} , only depending on its previous state, u_k , and no other states.

Suppose, however, scientists want to study the behavior of a red howler monkey in a more remote area. Knowing they will have limited opportunities of making visual observations, they attach a small microphone to one of the monkeys, whom they name Holly. The biologists discern that when hearing munching (M), the monkey is probably eating; but when hearing breathing (B) noises, the animal is likely to be at rest. Thus, the **set of possible observations** is $O = \{M, B\}$. These hypothetical researchers have developed a computer program to analyze the sounds and record B or M once an hour. However, the microphone/computational results are not completely accurate. Besides background noises, such as from rain or other monkeys, a sleeping Holly might be moving her mouth, perhaps dreaming of luscious leaves. For a while, the scientists are able to observe Holly personally and with their computer-enhanced microphone. In doing so, they discover that there is a 90% chance that if Holly is resting ($u_k = R$), then their monitoring system indicates breathing noises ($v_k = B$). (We select the symbol v to represent the “visible,” or observed, symbol obtained from the monitoring system). Thus, the **emission probability**, e , of B at state R is $e(B | R) = P(v_k = B | u_k = R) = 0.9$, so that $e(M | R) = P(v_k = M | u_k = R) = 1 - 0.9 = 0.1$. However, the scientists discover that the system is only 80% accurate in detecting eating; given that Holly is eating ($u_k = E$), their computer program interprets the audio signal as munching noises ($v_k = M$) 80% of the time. Thus, the emission probability of M given state E is $e(M | E) = P(v_k = M | u_k = E) = 0.8$ and $e(B | E) = P(v_k = B | u_k = E) = 1 - 0.8 = 0.2$. The HMM property of **output independence** states that regardless of the situation, the probability of an observation, v_k , only depends on the corresponding underlying state, u_k , that leads to the observation and no other states or observations. For example, the probability of the system displaying an output of munching (M) depends exclusively on underlying state of Holly eating (E).

After Holly scampers into the jungle, where the scientists cannot make visual observations, their monitoring system records an observation of B or M each hour. Using this **sequence of observations**, or **observed symbol sequence** ($v = v_1, v_2, v_3, \dots, v_n$, abbreviated $v_1 v_2 v_3 \dots v_n$), and the measures for initial, transition, and emission probabilities, (π, t, e) , the scientists can answer a number of questions. Such problems generally fall in one of three categories: likelihood, decoding, and learning problems. For example, the scientists might want to determine the probability, or likelihood, of obtaining an observed symbol sequence, $P(v)$, to discern if v is an unusual sequence of observations or not. As another **likelihood-type problem**, the scientists might want to determine the probability that a particular **sequence of states**, or **underlying state sequence** ($u = u_1 u_2 u_3 \dots u_n$) would generate a particular v ; so that they need to evaluate $P(v | u)$. Additionally, given a particular observation, v_k , the scientists might want to know the probability of an underlying state, u_k , written $P(u_k | v_k)$. Determining $P(u_k \neq u_{k+1} | v)$ represents a **change-detection problem**. In this case, given a sequence of monitoring system readings, we are determining the probability of Holly eating 1 h but resting the next, or vice versa. Perhaps in earlier studies we observed that usually the monkeys eat for exactly two time periods before sleeping deeply for at least 3 h, and we might want to use our system to estimate Holly's sleeping habits. In a **decoding-type problem**, they might be interested in finding the most likely u to generate a particular observed sequence, v . In a **learning-type problem** or **training-type problem**, for an observation sequence, v , and a set of states, we would be interested in determining the parameters for the system. For example, suppose we do not know the numbers for the initial state probabilities and in the transition and output matrices of the HMM associated with Holly. Given a long sequence of observations, such as $v = \text{MMBBBBBBMM} \dots \text{BB}$, a learning-type problem is to derive those numbers for the HMM that maximize the likelihood of observing the sequence, v . Determination of these parameters is called **training the HMM**.

In all cases, we are using observations and probabilities, which include a Markov model, to estimate something that is hidden. Hence, the name of this system is **Hidden Markov Model (HMM)**. The HMM for Holly consists in the following parameters, which Fig. 2, an expansion of Fig. 1, diagrams:

Holly's HMM

State space, or set of possible states, $S = \{E, R\}$, with elements representing eating and resting/sleeping, respectively.

Observation space, or set of possible observations, $O = \{M, B\}$, with elements representing munching and breathing noises, respectively.

Initial state probabilities, $\pi(E) = 0.30$ and $\pi(R) = 0.70$

Transition probabilities, $t(E, E) = 0.6$, $t(E, R) = 0.4$, $t(R, E) = 0.2$, and $t(R, R) = 0.8$, summarized by the following transition matrix:

(continued)

u_k/u_{k+1}	E	R
$T =$	E	$\begin{bmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{bmatrix}$

Emission probabilities, $e(M|E) = 0.8$, $e(B|E) = 0.2$, $e(M|R) = 0.1$, and $e(B|R) = 0.9$, summarized by the following output, or emission, matrix:

<i>hidden/observable</i>	M	B
E	$\begin{bmatrix} 0.8 & 0.2 \\ 0.1 & 0.9 \end{bmatrix}$	
R		

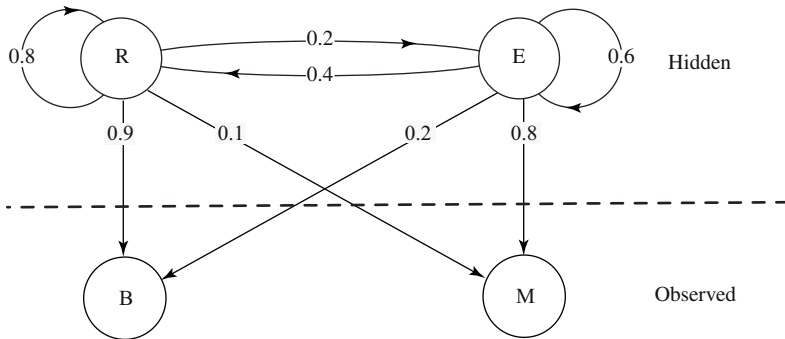


Fig. 2 Diagram of HMM for hypothetical study on red howler monkeys

Answers to Quick Review Questions appear at the end of the module, after “Projects.”

Quick Review Question 1

Consider the HMM diagram in Fig. 3 where the initial state probability for A is 0.2 and for B is 0.1. Determine each of the following:

- a. The set of possible states, S
- b. The set of possible observations, O
- c. $t(B, C)$
- d. $t(B, A)$
- e. The transition matrix with column headings being u_{k+1}
- f. $\pi(A)$
- g. $\pi(C)$
- h. $e(G|B)$

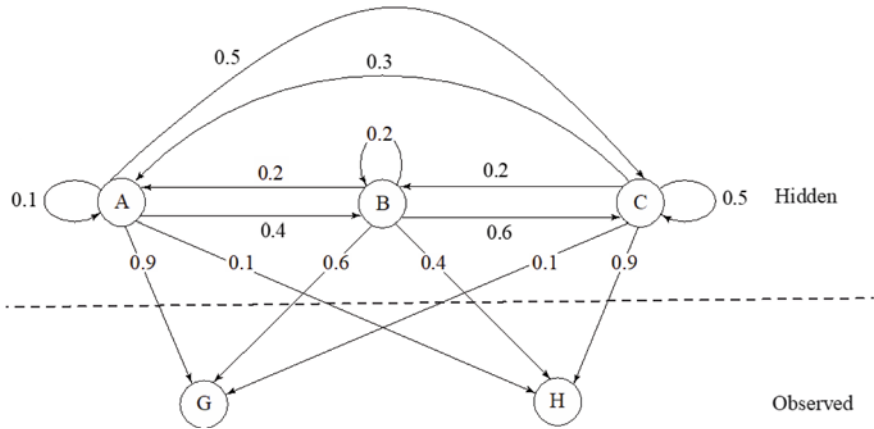


Fig. 3 HMM diagram for Quick Review Question 1

- i. $e(H | C)$
- j. The output (emission) matrix with column headings being observable values

3 Probability Equalities

Before we can start solving some of the problems with the HMM, we need to consider several probability equalities.

3.1 Joint Probability

Joint probability evaluates the probability of the simultaneous occurrence of two events, A and B , which are not necessarily independent as follows:

$$P(A \text{ and } B) = P(A, B) = P(A | B) \cdot P(B) \tag{1}$$

and

$$P(A \text{ and } B) = P(A, B) = P(B | A) \cdot P(A), \tag{2}$$

where the comma means “and,” or “intersection.”

For example, consider Holly’s HMM. Specifically, $P(E) = \pi(E) = 0.30$, $P(R) = \pi(R) = 0.70$, and the output, or emission, matrix is

$$\begin{array}{l}
 \text{hidden/observable} \quad M \quad B \\
 E \quad \quad \quad \left[\begin{array}{cc} 0.8 & 0.2 \end{array} \right] \\
 R \quad \quad \quad \left[\begin{array}{cc} 0.1 & 0.9 \end{array} \right]
 \end{array}$$

Knowing their probabilities, we can condition on the hidden states, E or R. For example, $P(\text{hidden} = E \text{ and } \text{observable} = M) = P(E, M) = P(M | E) \cdot P(E) = 0.8 \cdot 0.3 = 0.24$.

Quick Review Question 2

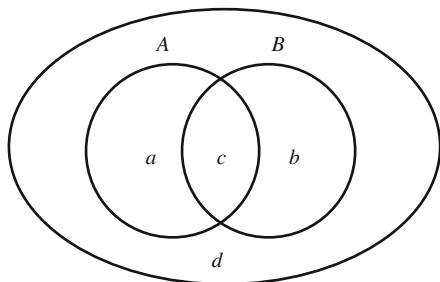
Calculate each of the following:

- a. $P(E, B)$
- b. $P(R, M)$
- c. $P(R, B)$
- d. $P(E, M) + P(E, B) + P(R, M) + P(R, B)$
- e. $\sum_{x \in \{M, B\}} P(E, x) = P(E, M) + P(E, B)$
- f. $\sum_{x \in \{M, B\}} P(R, x)$
- g. The sum of the answers to Parts e and f
- h. $\sum_{x \in \{E, R\}} P(x, M)$
- i. $\sum_{x \in \{E, R\}} P(x, B)$
- j. The sum of the answers to Parts h and i

To obtain an intuition for why the joint probability relationship Eq. (1) is true, consider the Venn diagram in Fig. 4, where the area for A consists in the areas a and c , while B contains areas B and c . Let us evaluate each term and verify that the left-hand side equals the right-hand side in Eq. (1). $P(A, B)$, the probability that an item is simultaneously in A and B , is the area of the intersection, c , divided by the diagram’s entire area, $(a + b + c + d)$, or

$$P(A, B) = \frac{c}{a + b + c + d}.$$

Fig. 4 Venn diagram



Asking for $P(A | B)$ means, “Given that an item is in B, what is the probability that the item is also in A?” B’s area is $b + c$, while the part of A that is simultaneously in B is c , so

$$P(A | B) = \frac{c}{b + c}.$$

The probability that an item is in B considers the area of B, $(b + c)$, in relationship to the whole diagram, whose area is $(a + b + c + d)$, so that

$$P(B) = \frac{b + c}{a + b + c + d}.$$

Substituting in Eq. (1) and simplifying, as follows, we verify that the left-hand side does indeed equal the right-hand side:

$$\begin{aligned} P(A, B) &\stackrel{?}{=} P(A|B) \cdot P(B) \\ \frac{c}{a + b + c + d} &\stackrel{?}{=} \frac{c}{\cancel{(b + c)}} \cdot \frac{\cancel{b + c}}{a + b + c + d} \\ \frac{c}{a + b + c + d} &= \frac{c}{a + b + c + d}. \end{aligned}$$

This argument is not a proof of Eq. (1) but does provide intuition into the equality.

Quick Review Question 3

Using a similar argument as for Eq. (1), provide an intuitive argument for Eq. (2).

If A and B are independent, so that A does not depend on B and vice versa, $P(A | B) = P(A)$ and $P(B | A) = P(B)$. Thus, for independent events, the equality reduces to the following:

$$P(A \text{ and } B) = P(A, B) = P(A) \cdot P(B) \text{ for independent } A \text{ and } B. \tag{3}$$

3.2 Marginal Probability

We can summarize the calculations from the last section, particularly Quick Review Question 2, for the joint probabilities in a matrix as follows:

	M	B	
E	0.24 0.06	0.30	(4)
R	0.07 0.63	0.70	
	0.31	0.69	1.00

Each matrix element gives a joint probability. For example, the probability that the monkey is eating (E) while the monitoring system simultaneously indicates munching (M) is $P(E, M) = 0.24$.

As developed in the quick review question, the sum of the probabilities in each row is to the right, and the sums of the column probabilities are below. When the monkey is eating (E), the equipment will either record munching (M) or breathing (B) with probabilities $P(E, M) = 0.24$ and $P(E, B) = 0.06$, respectively. Thus, the probability that the monkey is eating is $0.24 + 0.06 = 0.30$, a value indicated in Holly's HMM by $\pi(E) = 0.30$ and calculated in Quick Review Question 2 e for

$\sum_{x \in \{M, B\}} P(E, x) = P(E, M) + P(E, B)$. That is, the probability of E is the sum of the probabilities of hidden value E with each of the possible observable values, M and B; or $P(E) = \sum_{x \in \{M, B\}} P(E, x)$.

Additionally, the values below Matrix (4) are the sums of the probabilities in the columns. Here, we have new information. The probability of the equipment recording munching is 0.31; Holly is either eating (with probability 0.24) or resting/sleeping (with probability 0.07). Quick Review Question 2 h determined this value for $\sum_{x \in \{E, R\}} P(x, M) = P(E, M) + P(R, M)$, which is $P(M)$. We see that

the probability of observable value M is equal to the sum of the joint probabilities of M and every possible hidden state, E and R; or $P(M) = \sum_{x \in \{E, R\}} P(x, M)$.

These row and column sums are called **marginal probabilities** because they are written in the “margins” of the matrix. For example, the marginal probability of M is $P(M) = \sum_{x \in \{E, R\}} P(x, M) = P(E, M) + P(R, M) = 0.31$, and we are **marginalizing** over $\{E, R\}$. Notice also that the sum of the marginal probabilities on the right (or below) is 1.00, which is the sum of all the probabilities in the matrix. In general, the marginal probability of y is as follows:

$$P(y) = \sum_{x \in X} P(x, y) = \sum_{x \in X} P(y, x). \quad (5)$$

Quick Review Question 4

Consider the following matrix of joint probabilities:

$$\begin{array}{c} \text{F} \quad \text{G} \quad \text{H} \\ \text{J} \quad \left[\begin{array}{ccc} 0.05 & 0.18 & 0.20 \end{array} \right] \\ \text{K} \quad \left[\begin{array}{ccc} 0.36 & 0.11 & 0.10 \end{array} \right] \end{array}$$

- Determine $P(J)$.
- Determine the marginal probability of K.

- c. For Part b, give the set over which we are marginalizing.
- d. Using the matrix of joint probabilities, the answer for $P(J)$ in Part a, and joint probability formula (1) or (2), determine the conditional probability $P(G|J)$.
- e. Using the answer for Part b, determine $P(G|K)$.
- f. Write $P(K)$ using sigma notation.
- g. Determine $P(F)$.
- h. Using the answer for Part g, determine $P(J|F)$.
- i. Using the answer for Part g, determine $P(K|F)$.
- j. Determine the marginal probability of G.
- k. For Part j, give the set over which we are marginalizing.
 - l. Write $P(G)$ using sigma notation.
- m. Determine $P(H)$.

3.3 Bayes' Theorem

Bayes' Theorem, or **Bayes' Law**, which follows, is useful in manipulating conditional probabilities:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}. \quad (6)$$

Sometimes, we do not know $P(A|B)$ directly, but we do have values for each of the probability terms on the right. In this case, using Bayes' Theorem we can easily evaluate $P(A|B)$. For example, suppose we need to evaluate the probability of Holly eating (E) given that the equipment registers munching (M), $P(E|M)$. From Holly's HMM, we know $P(M|E) = e(M|E) = 0.8$ and $P(E) = \pi(E) = 0.30$. Moreover, in the "[Marginal Probability](#)" section, we calculated $P(M) = 0.31$. Thus, we can calculate $P(E|M)$ using Bayes' Law as follows:

$$P(E|M) = \frac{P(M|E) \cdot P(E)}{P(M)} = \frac{0.8 \cdot 0.3}{0.31} \approx 0.77.$$

Exercise 1 provides an intuitive justification for Bayes' Law.

Another version of Bayes' Theorem in which the comma means "and," or "intersection," is as follows:

$$P(A|B, C) = \frac{P(B|A, C) \cdot P(A|C)}{P(B|C)}. \quad (7)$$

Notice that this form is the same as the original version, except C is always in the condition. Using a Venn diagram, Exercise 2 provides intuition into Eq. (7).

Quick Review Question 5

Evaluate each of the following using $P(J) = 0.43$, $P(K) = 0.57$, $P(F) = 0.41$, $P(G) = 0.29$, $P(H) = 0.30$, $P(G|J) = 0.42$, $P(G|K) = 0.35$, $P(J|F) = 0.12$, $P(K|F) = 0.88$, and the matrix of joint probabilities from Quick Review Question 4:

$$\begin{array}{c} \text{F} \quad \text{G} \quad \text{H} \\ \text{J} \quad \left[\begin{array}{ccc} 0.05 & 0.18 & 0.20 \end{array} \right] \\ \text{K} \quad \left[\begin{array}{ccc} 0.36 & 0.11 & 0.10 \end{array} \right] \end{array}$$

- a. $P(J|G)$
- b. $P(K|G)$
- c. $P(F|J)$
- d. $P(F|K)$

4 Probability of a State Given an Observation

Using Bayes' Theorem, other probability equalities, and HMM, we can determine the probabilities of a variety of situations. For example, suppose after Holly scampers off into the jungle, at hour k the monitoring equipment registers $v_k = M$, munching. The scientists might wonder if she really is eating, or, symbolically, if it is true that $u_k = E$. They are asking, "For this single munching reading and without reference to earlier readings, what is the likelihood, or probability, that Holly is eating?" This question falls in the category of an HMM likelihood problem. In notation, they want to discover $P(u_k = E | v_k = M)$, abbreviated $P(E | M)$; that is, given a reading of M, what is the probability that Holly is in state E.

Examining the HMM model, we do not find the answer directly. However, using Bayes' Theorem, we can rewrite the question as follows:

$$P(u_k = E | v_k = M) = \frac{P(v_k = M | u_k = E) \cdot P(u_k = E)}{P(v_k = M)}$$

or, alternatively,

$$P(E | M) = \frac{P(M | E) \cdot P(E)}{P(M)}.$$

The advantage of this equality is that we can evaluate each of the terms on the right. Consulting Holly's HMM, $P(M | E) = e(M | E) = 0.8$, and $P(E) = \pi(E) = 0.30$. The denominator, $P(M)$ does take a bit more thought. If the monitoring system records $v_k = M$, Holly could either be eating ($u_k = E$, $v_k = M$) or resting ($u_k =$

R, $v_k = M$). Thus, marginalizing over E, R, we have

$$P(v_k = M) = P(u_k = E, v_k = M) + P(u_k = R, v_k = M)$$

or

$$P(M) = P(E, M) + P(R, M).$$

Using a joint probability to determine the first summand, we have

$$P(E, M) = P(M, E) = P(M | E) \cdot P(E).$$

That is, the probability of Holly eating leaves and the system recording munching is the same as the probability that the equipment correctly records munching when Holly is eating *and* Holly really is enjoying her dinner. Similarly,

$$P(R, M) = P(M | R) \cdot P(R).$$

The probability of Holly resting and the system indicating munching is identical to the probability of an incorrect recording of munching when she is resting and Holly is actually inactive. Thus, putting the pieces together determined from joint and marginal probabilities, we have

$$\begin{aligned} P(M) &= P(E, M) + P(R, M) \\ &= P(M | E) \cdot P(E) + P(M | R) \cdot P(R). \end{aligned}$$

Fortunately, from Holly’s HMM, we know each of the terms on the right: $P(M | E) = e(M | E) = 0.8$, $P(E) = \pi(E) = 0.30$, $P(M | R) = e(M | R) = 0.1$, and $P(R) = \pi(R) = 0.70$. Incorporating all calculations, we have the following:

$$\begin{aligned} P(E | M) &= \frac{P(M | E) \cdot P(E)}{P(M)} = \frac{P(M | E) \cdot P(E)}{P(M | E) \cdot P(E) + P(M | R) \cdot P(R)} \\ P(E | M) &= \frac{0.8 \cdot 0.3}{0.8 \cdot 0.3 + 0.1 \cdot 0.7} \approx 0.77. \end{aligned}$$

There is a 77% chance of Holly eating when the system records munching.

In general, for Holly’s situation, the following equation determines the probability of an underlying state, u_k , given an observation, v_k :

$$P(u_k | v_k) = \frac{P(v_k | u_k) \cdot P(u_k)}{P(v_k)} = \frac{P(v_k | u_k) \cdot P(u_k)}{P(v_k | E) \cdot P(E) + P(v_k | R) \cdot P(R)}$$

or, using sigma notation,

$$P(u_k | v_k) = \frac{P(v_k | u_k) \cdot P(u_k)}{\sum_{x \in \{E, R\}} P(v_k | x) \cdot P(x)}.$$

However, most systems have more than two states. Thus, for other HMMs with S being the set of all (hidden) states, we have the following:

$$P(u_k | v_k) = \frac{P(v_k | u_k) \cdot P(u_k)}{\sum_{x \in S} P(v_k | x) \cdot P(x)}. \tag{8}$$

Quick Review Question 6

For Holly’s HMM, determine the following:

- a. $P(R | M)$
- b. $P(R | B)$
- c. $P(E | B)$

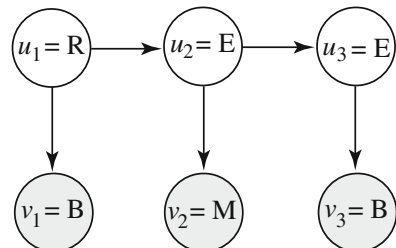
Quick Review Question 7

Suppose $P(F) = 0.41$, $P(G) = 0.29$, $P(H) = 0.30$, $P(J | F) = 0.12$, $P(J | G) = 0.62$, $P(J | H) = 0.26$. Determine $P(F | J)$.

5 Probability of a Sequence of States Generating a Sequence of Observations

In this section, we consider a problem that would usually not occur but whose answer will help us in the solution of other more realistic problems: Knowing a sequence of states, u , what is the probability of a particular sequence of observations, v , or $P(v | u)$. For example, suppose we could spy on Holly in the jungle and discover that initially she rested ($u_1 = R$), but in the next 2 h she ate ($u_2 = E$ and $u_3 = E$), so that $u = REE$. Also, suppose we want to discover the probability that the monitoring equipment registers breathing ($v_1 = B$) followed by munching and then breathing noises in the next two readings ($v_2 = M$ and $v_3 = B$), or $v = BMB$. Thus, we are interested in $P(v = BMB | u = REE) = P(BMB | REE)$. Figure 5 presents a **trellis diagram** of the situation, with state circles unshaded, observation circles shaded, and arrows denoting conditional dependencies.

Fig. 5 Trellis diagram with state circles unshaded, observation circles shaded, and arrows denoting conditional dependencies



Because of the HMM property of output independence, the probability of each observation, v_i , only depends on the corresponding state, u_i . For example, the probability that the equipment's third reading is B, given that Holly is eating the third hour, $P(v_3 = B | u_3 = E)$, is the emission probability, $e(B | E) = 0.2$. Because of independence, to evaluate $P(\text{BMB} | \text{REE})$, we take the product of the three corresponding emission probabilities as follows:

$$\begin{aligned} P(\text{BMB} | \text{REE}) &= P(B | R) \cdot P(M | E) \cdot P(B | E) \\ &= e(B | R) \cdot e(M | E) \cdot e(B | E) \\ &= 0.9 \cdot 0.8 \cdot 0.2 \\ &= 0.144. \end{aligned}$$

In general, for sequences of length n , $u = u_1 u_2 \dots u_n$ and $v = v_1 v_2 \dots v_n$, we have

$$P(v|u) = e(v_1|u_1) \cdot e(v_2|u_2) \dots e(v_n|u_n).$$

We can abbreviate the product with pi notation to yield the following equality for the probability of an observation sequence given a hidden sequence:

$$P(v | u) = \prod_{i=1}^n e(v_i | u_i). \quad (9)$$

Quick Review Question 8

- a. Evaluate $P(\text{MMMB} | \text{ERER})$
- b. Suppose a team catches up with Holly in the jungle and finds her sleeping for 6 h. Calculate the probability that the equipment in the base camp registers correctly records breathing over that period.

6 Probability of a State Sequence and an Observation Sequence

Another likelihood problem considers the probability of a particular observation sequence and a particular state sequence. There is an important difference between this and the problem in the previous section. For the previous problem, given a hidden sequence, u , we wanted to determine the conditional probability of a particular visible sequence, v , namely $P(v | u)$. In the current problem, we desire the joint probability that both a hidden sequence, u , and a visible sequence, v , occur, or $P(v, u)$.

For example, in the case of Holly, we might want to know the chance of the equipment registering a sequence of breathing (B), munching (M), and breathing (B), while she is actually resting (R), eating (E), and eating (E). By the joint probability equality (1), we can calculate this probability, $P(\text{BMB}, \text{REE})$, as follows:

$$P(\text{BMB}, \text{REE}) = P(\text{BMB} | \text{REE}) \cdot P(\text{REE}). \quad (10)$$

That is, the joint probability of both BMB and REE is the probability of BMB given REE and that REE really does occur.

We determined the conditional probability $P(\text{BMB} | \text{REE})$ in the previous section as 0.144, while Holly's Markov model enables us to calculate the chance of the hidden state sequence REE. As the trellis diagram in Fig. 5 illustrates, REE involves three independent events: Holly is initially resting; given that she is resting the first hour, she is eating the next hour; and she continues eating from the second hour to the next. Thus, the probability of the state sequence REE is as follows:

$$P(\text{REE}) = P(u_1 = \text{R}) \cdot P(u_2 = \text{E} | u_1 = \text{R}) \cdot P(u_3 = \text{E} | u_2 = \text{E}).$$

From Holly's HMM, we know $P(u_1 = \text{R}) = \pi(\text{R}) = 0.70$ and the transition probabilities $P(u_2 = \text{E} | u_1 = \text{R}) = t(\text{R}, \text{E}) = 0.2$ and $P(u_3 = \text{E} | u_2 = \text{E}) = t(\text{E}, \text{E}) = 0.6$. Thus, the probability of the state sequence REE is as follows:

$$P(\text{REE}) = \pi(\text{R}) \cdot t(\text{R}, \text{E}) \cdot t(\text{E}, \text{E}) = 0.70 \cdot 0.2 \cdot 0.6 = 0.084.$$

Substituting values into Eq. (10), we have the following:

$$P(\text{BMB}, \text{REE}) = P(\text{BMB} | \text{REE}) \cdot P(\text{REE}) = 0.144 \cdot 0.084 \approx 0.012.$$

Although given hidden sequence REE, there is approximately a 14% chance of observable sequence BMB, there is only about a 1% chance of BMB and REE simultaneously occurring.

Quick Review Question 9

Use the answers from Quick Review Question 8 as needed.

- a. Evaluate $P(\text{MMMM}, \text{ERER})$
- b. Find the probability that Holly sleeps for 6 h and the monitoring equipment registers breathing over that same time period.

Generalizing, based on the Markov property, the probability of a sequence of observations, u , is as follows:

$$P(u) = \pi(u_1) \cdot \prod_{i=2}^n t(u_{i-1}, u_i). \quad (11)$$

Moreover, considering individual sequence elements, we have the following:

$$P(v | u) = \prod_{i=1}^n P(v_i | u_i) = \prod_{i=1}^n e(v_i | u_i). \tag{12}$$

Thus, using Eqs. (11) and (12), the generalized formula for the joint probability of particular observable and hidden sequences is as follows:

$$P(v, u) = P(v | u) \cdot P(u) = \left(\prod_{i=1}^n e(v_i | u_i) \right) \left(\pi(u_1) \cdot \prod_{i=2}^n t(u_{i-1}, u_i) \right). \tag{13}$$

A useful reorganization of Eq. (13), which follows, is evocative of the trellis diagram in Fig. 5:

$$\begin{aligned} P(v, u) &= \pi(u_1) \cdot e(v_1 | u_1) \cdot \left(\prod_{i=2}^n [t(u_{i-1}, u_i) \cdot e(v_i | u_i)] \right) \\ &= \pi(u_1) \cdot e(v_1 | u_1) \cdot t(u_1, u_2) \cdot e(v_2 | u_2) \cdot t(u_2, u_3) \cdot \\ &\quad e(v_3 | u_3) \cdots t(u_{n-1}, u_n) \cdot e(v_n | u_n). \end{aligned}$$

Regrouping this expression, as follows, is revealing for the joint probability $P(v, u) = P(u, v)$:

$$P(v, u) = \left[\dots \left[\left[\pi(u_1) \cdot e(v_1 | u_1) \right] \cdot t(u_1, u_2) \cdot e(v_2 | u_2) \right] \cdot t(u_2, u_3) \cdot e(v_3 | u_3) \right] \cdots \cdot t(u_{n-1}, u_n) \cdot e(v_n | u_n).$$

Notice that the product of the first two terms, which are in the inner brackets, $[\pi(u_1) \cdot e(v_1 | u_1)]$, is $P(u_1, v_1)$, or $P(u, v)$, when u and v are one-element sequences. Moreover, the product of the terms in the next set of brackets is $P(u_{1,2}, v_{1,2})$, which is $P(u, v)$ when u and v are two-element sequences. The expression in the next set of brackets calculates $P(u_{1,3}, v_{1,3})$; and the expression in the outermost set of brackets is $P(u_{1,n-1}, v_{1,n-1})$, which is $P(u, v)$ for sequences of $n - 1$ elements. Thus, we can use recursion to define $P(u, v)$. A recursive task is one that calls itself. In this example, we employ the joint probability $P(u_{1,n-1}, v_{1,n-1})$ in the definition of the joint probability $P(u, v)$. The following recursive formula for $P(u, v)$ will be useful in further algorithms:

$$P(u, v) = P(u_{1,n}, v_{1,n}) = \begin{cases} \pi(u_1) \cdot e(v_1 | u_1), & \text{if } n = 1 \\ P(u_{1,n-1}, v_{1,n-1}) \cdot t(u_{n-1}, u_n) \cdot e(v_n | u_n), & \text{if } n > 1 \end{cases} \tag{14}$$

7 Probability of a Sequence of Observations: The Forward Algorithm

A more realistic likelihood problem is to determine the probability of an observed sequence, $P(v)$. For example, we might want to calculate the probability that the monitoring equipment registers breathing sounds and then munching noises the next 2 h, $P(v = \text{BMM})$.

7.1 Obvious Solution

In the last section, we determined the simultaneous probability of a visible and a hidden sequence. Thus, the obvious solution for this short sequence is to marginalize over all the possible three-element hidden state sequences. For $S = \{E, R\}$, there are two choices, E and R, for each of the three positions, yielding $2^3 = 8$ possible hidden sequences. Thus, marginalizing over this set of eight underlying sequences, $U = \{\text{RRR}, \text{RRE}, \text{RER}, \text{REE}, \text{ERR}, \text{ERE}, \text{EER}, \text{EEE}\}$, we can calculate $P(\text{BMM})$ as follows:

$$P(\text{BMM}) = \sum_{u \in U} P(\text{BMM}, u) = \sum_{u \in U} [P(\text{BMM} | u) \cdot P(u)].$$

Expanding the first sum, we have the following:

$$\begin{aligned} P(\text{BMM}) &= P(\text{BMM}, \text{RRR}) + P(\text{BMM}, \text{RRE}) + P(\text{BMM}, \text{RER}) \\ &\quad + P(\text{BMM}, \text{REE}) + P(\text{BMM}, \text{ERR}) + P(\text{BMM}, \text{ERE}) \\ &\quad + P(\text{BMM}, \text{EER}) + P(\text{BMM}, \text{EEE}). \end{aligned}$$

We can calculate each of these joint probabilities using Eqs. (12) or (13) from the last section. Such a solution seems feasible, although tedious. However, consider the situation where we have sequences of 10 elements. The number of 10-element sequences formed from $\{E, R\}$ is $2^{10} = 1024$, so that we would have 1024 summands. Doubling to 20 observations, the number of possible sequences, and, hence, the number of summands, is over a million, $2^{20} = 1,048,576$. Astoundingly, the number sequences of length 100 is over 10^{30} . As these calculations illustrate, for h hidden states and n observations, there are h^n number of possible hidden sequences; as n gets larger, h^n grows exponentially. Thus, this solution has an exponential growth rate on the order of h^n , written $O(h^n)$.

Quick Review Question 10

For each string length, find the number of strings formed from the bases A, C, T, and G.

- a. 3
- b. 10
- c. 20
- d. 21

7.2 Forward Algorithm

For realistic problems, often the number of states (h) and certainly the number of observations (n) are large, so that h^n is enormous, too great to compute by hand, and, in fact, intractable even for a computer. Clearly, we must find a better way to calculate $P(v)$, the probability of an observed sequence. Fortunately, the **forward algorithm** is much faster. The algorithm employs **dynamic programming**, which divides a problem into a collection of smaller problems and uses the solutions to these smaller problems to solve the larger problem. For the forward algorithm, we store answers to the smallest problems in the first column of a matrix. Repeatedly, we solve progressively larger problems, employing the answers in the previous column and storing the answers in a new column. Finally, the answer to the overall problem, the probability of an observed sequence, is the sum of the elements in the last column.

For $P(\text{BMM})$, we employ a matrix with the same number of rows as the number of states, h , and the same number of columns as the length of the observation sequence, n . For Holly’s HMM, with $S = \{E, R\}$ and observed sequence $v = \text{BMM}$, $h = 2$ and $n = 3$, so we store values in a 2×3 matrix, F , with row and column headings from S and v , respectively:

$$F = \begin{matrix} & & \text{B} & \text{M} & \text{M} \\ \begin{matrix} \text{E} \\ \text{R} \end{matrix} & \begin{bmatrix} \square & \square & \square \\ \square & \square & \square \end{bmatrix} \end{matrix}.$$

7.3 Forward Algorithm Initialization

Initially, we solve the smaller problem of $P(\text{B})$, which by marginalizing Eq. (5) is as follows:

$$P(\text{B}) = \sum_{x \in S} P(\text{B}, x) = P(\text{B}, \text{E}) + P(\text{B}, \text{R}).$$

We will place the calculation of $P(\text{B}, \text{E})$ in $f_{\text{E}1}$, the element in the E row and first column of F, while $f_{\text{R}1}$ will be $P(\text{B}, \text{R})$. The probability of $u_1 = \text{B}$ is the sum of these two values in the first column. For the calculation of the summands, we employ the step for $n = 1$ of the recursive formula (14) as follows:

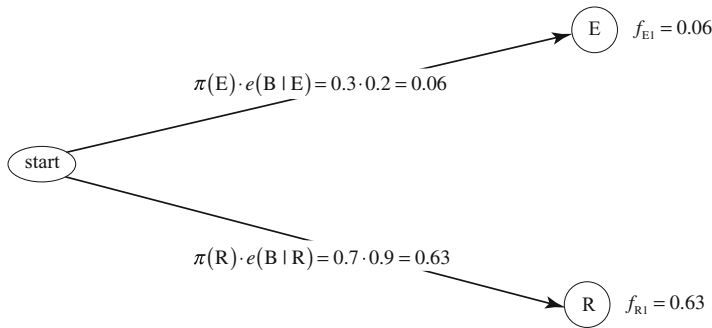


Fig. 6 Initialization step of the forward algorithm

$$P(B, E) = \pi(E) \cdot e(B | E)$$

and

$$P(B, R) = \pi(R) \cdot e(B | R).$$

The initial probabilities are $\pi(E) = 0.30$ and $\pi(R) = 0.70$, and the emission probabilities are $e(B | E) = 0.2$, $e(B | R) = 0.9$. Thus, as Fig. 6 illustrates, $P(B, E) = 0.06$ and $P(B, R) = 0.63$. Thus, placing these values from the initialization step in the first column, the matrix F is as follows:

$$F = \begin{matrix} & \begin{matrix} B & M & M \end{matrix} \\ \begin{matrix} E \\ R \end{matrix} & \begin{bmatrix} 0.06 & \square & \square \\ 0.63 & \square & \square \end{bmatrix} \end{matrix}.$$

Quick Review Question 11

The HMM in Quick Review Question 1 and Fig. 3 contains the following information:

$S = \{A, B, C\}$ and $O = \{G, H\}$

$\pi(A) = 0.2$, $\pi(B) = 0.1$, $\pi(C) = 0.7$

$$T = \begin{matrix} u_k / u_{k+1} & \begin{matrix} A & B & C \end{matrix} \\ \begin{matrix} A \\ B \\ C \end{matrix} & \begin{bmatrix} 0.1 & 0.4 & 0.5 \\ 0.2 & 0.2 & 0.6 \\ 0.3 & 0.2 & 0.5 \end{bmatrix} \end{matrix}$$

<i>hidden/observable</i>	G	H
A	$\begin{bmatrix} 0.9 & 0.1 \\ 0.6 & 0.4 \\ 0.1 & 0.9 \end{bmatrix}$	
B		
C		

Suppose we wish to use the forward algorithm to calculate $P(\text{HHGH})$.

- a. Give the size of F , the forward algorithm matrix.
- b. Calculate f_{A1} .
- c. Write the conditional probability for f_{A1} symbolically; that is, write f_{A1} as $P(x | y)$ with the symbols for x and y for this example.
- d. Calculate f_{B1} .
- e. Write the conditional probability for f_{B1} symbolically.
- f. Calculate f_{C1} .
- g. Write the conditional probability for f_{C1} symbolically.
- h. Calculate $P(\text{H})$.

7.4 Forward Algorithm Step 2

In the calculation of $P(\text{BMM})$, after initialization, which calculates the summands of $P(\text{B})$, we approach a little bit larger problem, $P(\text{BM})$. Fortunately, as we will see, we can use the values in the first column in the solution of this probability. As we did in the initialization step, we employ marginality to split the problem into a sum of joint probabilities. With marginality, we have

$$P(\text{BM}) = \sum_{x \in S^2} P(\text{BM}, x), \tag{15}$$

where S^2 represents the set of all two-element sequences with elements from $S = \{\text{E}, \text{R}\}$, $S^2 = \{\text{EE}, \text{ER}, \text{RE}, \text{RR}\}$; and, in general, S^n denotes the set of all n -element sequences over S . Thus, expanding Eq. (15) symbolically, we have the following sum:

$$P(\text{BM}) = P(\text{BM}, \underline{\text{EE}}) + P(\text{BM}, \underline{\text{ER}}) + P(\text{BM}, \underline{\text{RE}}) + P(\text{BM}, \underline{\text{RR}}). \tag{16}$$

Let us calculate one of these summands, $P(\text{BM}, \text{ER})$. Applying the second line of the recursive definition of $P(u, v)$, Eq. (14), we have the following:

$$P(\text{BM}, \text{ER}) = P(\text{B}, \text{E}) \cdot t(\text{E}, \text{R}) \cdot e(\text{M} | \text{R}).$$

Fortunately, we calculated $P(B,E)$ in the initialization step, and its value is matrix element $f_{E1} = 0.06$. Moreover, Holly’s HMM gives $t(E, R)$ and $e(M | R)$ as 0.4 and 0.1, respectively. Thus, $P(BM, ER) = 0.06 \cdot 0.4 \cdot 0.1 = 0.0024$.

Quick Review Question 12

Calculate $P(BM, RR)$.

For f_{E2} we add the elements of Eq. (16) with sequences that end in E. Similarly, f_{R2} is the sum of elements with sequences that end in R. Rewriting Eq. (16) illustrates the process:

$$\begin{aligned}
 P(BM) &= [P(BM, \underline{EE}) + P(BM, \underline{RE})] + [P(BM, \underline{ER}) + P(BM, \underline{RR})] \\
 &= f_{E2} + f_{R2}.
 \end{aligned}
 \tag{17}$$

With sigma notation, we have the following equations for the second-column elements:

$$f_{E2} = \sum_{x \in S} P(BM, xE)$$

and

$$f_{R2} = \sum_{x \in S} P(BM, xR).$$

From the calculations above and in Quick Review Question 12, we know that $P(BM, ER) = 0.0024$ and $P(BM, RR) = 0.0504$. Thus, $f_{R2} = P(BM, ER) + P(BM, RR) = 0.0024 + 0.0504 = 0.0528$.

Quick Review Question 13

Using $P(BM, EE) = 0.0288$ and $P(BM, RE) = 0.1008$, calculate f_{R2} .

Figure 7 illustrates the calculation of the second column of matrix F with $P(BM, EE) + P(BM, RE) = f_{E2}$ and $P(BM, ER) + P(BM, RR) = f_{R2}$. Thus, the developing F now is as follows:

$$\begin{array}{cc}
 & \begin{matrix} B & M & M \end{matrix} \\
 \begin{matrix} E \\ R \end{matrix} & \begin{bmatrix} 0.06 & 0.1296 & \square \\ 0.63 & 0.0528 & \square \end{bmatrix}.
 \end{array}$$

Moreover, the sum of the elements in the second column, 0.1824, is $P(BM)$; that is, for sequences of length two, there is an 18.24% chance of BM being an output sequence.

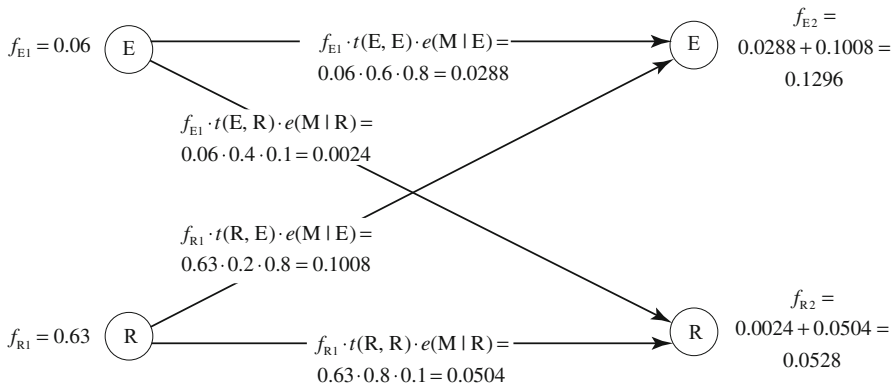


Fig. 7 Calculation of the second column of forward matrix, F

Quick Review Question 14

The HMM in Quick Review Questions 1 and 11 and Fig. 3 contains the following information:

$S = \{A, B, C\}$ and $O = \{G, H\}$
 $\pi(A) = 0.2, \pi(B) = 0.1, \pi(C) = 0.7$

$$T = \begin{matrix} u_k/u_{k+1} & A & B & C \\ A & \begin{bmatrix} 0.1 & 0.4 & 0.5 \end{bmatrix} \\ B & \begin{bmatrix} 0.2 & 0.2 & 0.6 \end{bmatrix} \\ C & \begin{bmatrix} 0.3 & 0.2 & 0.5 \end{bmatrix} \end{matrix}$$

$$\begin{matrix} hidden/observable & G & H \\ A & \begin{bmatrix} 0.9 & 0.1 \end{bmatrix} \\ B & \begin{bmatrix} 0.6 & 0.4 \end{bmatrix} \\ C & \begin{bmatrix} 0.1 & 0.9 \end{bmatrix} \end{matrix} .$$

Suppose we wish to use the forward algorithm to calculate $P(HHGH)$. As calculated in Quick Review Question 11, the first column of the forward algorithm matrix F contains $f_{A1} = 0.02, f_{B1} = 0.04,$ and $f_{C1} = 0.63$.

- Give the number of summands in the calculation of f_{B2} .
- For f_{B2} , calculate the first summand, which employs f_{A1} as a factor.
- For f_{B2} , calculate the second summand, which employs f_{B1} as a factor.
- For f_{B2} , calculate the third summand.
- Calculate f_{B2} .

7.5 Forward Algorithm Completion

The calculation of the third column of F for $P(\text{BMM})$ proceeds in a similar manner to that of the second column. We use the small-problem answers in the previous column to calculate the values for the new column. Quick Review Question 15 steps through the process.

Quick Review Question 15

This question completes the forward algorithm for $P(\text{BMM})$.

- a. Write $P(\text{BMM})$ using sigma notation similar to that in Eq. (15).
- b. Write out S^3 .
- c. Enumerate $P(\text{BMM})$ symbolically as in Eq. (17), grouping together the elements of f_{E3} and f_{R3} .
- d. Notice in the calculation of f_{E3} that two probabilities have sequences that end in EE, $P(\text{BMM}, \text{EEE})$ and $P(\text{BMM}, \text{REE})$. Moreover, by Eq. (14), we have the following:

$$P(\text{BMM}, \text{EEE}) = P(\text{BM}, \text{EE}) \cdot t(\mathbf{E}, \mathbf{E}) \cdot e(\mathbf{M} | \mathbf{E})$$

$$P(\text{BMM}, \text{REE}) = P(\text{BM}, \text{RE}) \cdot t(\mathbf{E}, \mathbf{E}) \cdot e(\mathbf{M} | \mathbf{E}).$$

With the last two factors being identical, we can group the sum of the probabilities as follows:

$$P(\text{BMM}, \text{EEE}) + P(\text{BMM}, \text{REE}) = [P(\text{BM}, \text{EE}) + P(\text{BM}, \text{RE})] \cdot t(\mathbf{E}, \mathbf{E}) \cdot e(\mathbf{M} | \mathbf{E}).$$

In a similar manner, show the development to write the sum of the remaining two terms of f_{E3} using f_{R2} .

- e. As in Part d, write the sum of two terms of f_{E3} using f_{E2} .
- f. As in Part d, write the sum of two terms of f_{R3} using f_{E2} .
- g. Evaluate $P(\text{BMM}, \text{EEE}) + P(\text{BMM}, \text{REE})$, developed symbolically in Part d.
- h. Referring to the answer to Part d, evaluate $P(\text{BMM}, \text{ERE}) + P(\text{BMM}, \text{RRE})$.
- i. Referring to the answer to Part e, evaluate $P(\text{BMM}, \text{EER}) + P(\text{BMM}, \text{RER})$.
- j. Referring to the answer to Part f, evaluate $P(\text{BMM}, \text{ERR}) + P(\text{BMM}, \text{RRR})$.
- k. Using the answers to Parts g and i, calculate f_{E3} .
- l. Using the answers to Parts h and j, calculate f_{R3} .
- m. Calculate $P(\text{BMM})$

Figure 8 illustrates the development of the last column of the forward matrix, which Quick Review Question (15) solved. Moreover, with this and previous calculations, we can complete the forward matrix as follows:

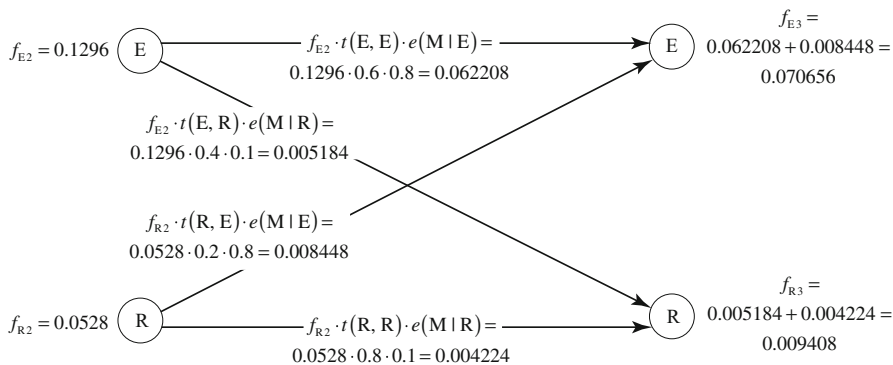


Fig. 8 Step 3 of forward algorithm in calculation of $P(\text{BMM})$

$$F = \begin{matrix} & \text{B} & \text{M} & \text{M} \\ \text{E} & \begin{bmatrix} 0.06 & 0.1296 & 0.070656 \\ 0.63 & 0.0528 & 0.094080 \end{bmatrix} \\ \text{R} & \end{matrix} \tag{18}$$

Then, we can calculate $P(\text{BMM})$ as the sum of the elements in the last column, or $P(\text{BMM}) = 0.080064$. There is approximately an 8% chance of the output sequence BMM.

As the work for Quick Review Question 15 indicates, we can iterate through the matrix, using one column in the evaluation of the next. For $S = \{E, R\}$ and i th output symbol, v_i , we have:

$$\begin{aligned} f_{Ei} &= f_{E(i-1)} \cdot t(E, E) \cdot e(v_i | E) + f_{R(i-1)} \cdot t(R, E) \cdot e(v_i | E) \\ &= \sum_{y \in S} [f_{y(i-1)} \cdot t(y, E) \cdot e(v_i | E)] \end{aligned}$$

and

$$\begin{aligned} f_{Ri} &= f_{E(i-1)} \cdot t(E, R) \cdot e(v_i | R) + f_{R(i-1)} \cdot t(R, R) \cdot e(v_i | R) \\ &= \sum_{y \in S} [f_{y(i-1)} \cdot t(y, R) \cdot e(v_i | R)]. \end{aligned}$$

In general, for state x ,

$$f_{xi} = \sum_{y \in S} [f_{y(i-1)} \cdot t(y, x) \cdot e(v_i | x)].$$

Moreover, the sum of the elements in column i is the probability of the sequence of the first i observations, $P(v_{1,i})$. For example, in (18), we completed the forward matrix for $P(\text{BMM})$ to be the following:

$$F = \begin{array}{c} \text{E} \\ \text{R} \end{array} \begin{array}{ccc} \text{B} & \text{M} & \text{M} \\ \left[\begin{array}{ccc} 0.06 & 0.1296 & 0.070656 \\ 0.63 & 0.0528 & 0.094080 \end{array} \right] \cdot \end{array}$$

The sum of the elements in the first column is $P(\text{B}) = 0.69$, while the sum of the second-column elements is $P(\text{BM}) = 0.1824$. The answer we desire, $P(\text{BMM})$, is the sum of the elements in the final column, 0.080064; the probability of BMM is approximately 8%. To summarize,

$$P(v) = \sum_{x \in S} f_{xn}, \quad (19)$$

where v is a sequence of n observations and f_{xn} is the element in row x and column n of the forward matrix F . The following quick review question considers the total number of calculations for such a probability using the forward algorithm.

Quick Review Question 16

This question analyzes the complexity of the forward algorithm.

- Suppose a HMM has 5 hidden states. Give the number of rows in the forward matrix, F .
- In Holly's HMM with 2 hidden states, the calculation of each of the two elements in the first column involved one product, for a total of 2 products. For 5 hidden states, give the total number of products in the evaluation of the first column.
- For h states, give the total number of products in the evaluation of the first column.
- Suppose x is one of these 5 hidden states. Not including the initialization step, give the number of terms having $f_{x(i-1)}$ as a factor, where $f_{x(i-1)}$ is the forward matrix value in row x and column $(i-1)$. This value is the same number of lines emanating from an f -value. For example, in Fig. 8, two terms had f_{2E} (or f_{2R}) as a factor and two lines emanating from f_{2E} (or f_{2R}).
- For 2 hidden states, each of these terms involves two products. Give the number of products when we have 5 hidden states.
- Give the number of these products when we have h hidden states. Note that the number of products does not depend on the number of hidden states.
- Give the number of elements in the forward matrix for 5 hidden states and 20 observations, not including those in the first column.
- Give the number of elements in the forward matrix for h hidden states and n observations, not including those in the first column.
- Using your answers from Parts b, e, and g, for 5 hidden states and 20 observations, give the total number of products.

- j. Using your answers from Parts c, f, and h, for h hidden states and n observations, give the total number of products. Simplify the result.
- k. Give the total number of sums used in the calculations of the elements in the first column of a forward matrix, F .
- l. For 5 hidden states and not including the initialization step, give the number of sums used in the calculation of f_{xi} , where x is a hidden state and i is a column. For example, in Fig. 8, f_{3E} (or f_{3R}) has two summands and, thus, one sum.
- m. For h hidden states, give the number of sums used in the calculation of f_{xi} , where x is a hidden state and i is a column greater than 1.
- n. For h hidden states and n observations, give the total number of sums used in the calculations of the elements of the forward matrix. Use answers from Parts h, k, and m. Simplify the result.
- o. For h hidden states and n observations, give the total number of sums and products in the calculation of the elements of F . Use answers from Parts j and n.
- p. Give the complexity of the forward algorithm.

As indicated in Quick Review Question 16, for h states and n observations, the forward matrix, F , has h rows and n columns, yielding hn number of elements. Most of these elements employ h summands of three factors each, although elements in the first column have fewer terms. Thus, the approximate amount of work involved in creating the forward matrix is on the order of nh^2 . Moreover, to calculate the probability of the observed sequence, we add the h elements in the last column. In comparison to nh^2 , h is inconsequential, so the forward algorithm is on the order of nh^2 , written $O(nh^2)$. The complexity $O(nh^2)$ is a dramatic improvement over the exponential complexity, $O(h^n)$, for our original solution. For example, suppose our HMM has $h = 10$ hidden states and we wish to know the probability of a sequence of $n = 9$ observations. The original approach to the solution would involve $h^n = 10^9 = 1,000,000,000$, a billion calculations, while the forward algorithm would employ only about $nh^2 = 9 \cdot 10^2 = 900$ calculations.

8 Probability of a Genomics Sequence

In this section, we employ the forward algorithm to solving an important biological problem—locating genes.

8.1 Biology Background

Proteins, which are the basic molecules of life, perform many essential functions, such as forming the structural components of cells and, in the case of protein enzymes, catalyzing chemical reactions. Simple proteins are chains formed from 20 amino acids.

In the cell, the nucleic acid **DNA (deoxyribonucleic acid)** contains the encoded information for the building of all the proteins needed by a cell. DNA acts through an intermediary nucleic acid, **RNA (ribonucleic acid)**, to synthesize proteins. RNA has one long chain of molecules, called **nucleotides**, while two strands of nucleotides compose DNA. Each nucleotide is made up of a sugar, a phosphate, and a nitrogen base, which can be **adenine (A)**, **guanine (G)**, **cytosine (C)**, or **thymine (T)** in DNA or **uracil (U)** in RNA. In DNA, A in one strand bonds with T in the other strand, while C and G bond together. Each of these pairs of **complementary bases** is referred to as a **base pair (bp)**.

Virtually every cell (except red blood cells) in the human body contains **chromosomes**, or sequences of very long DNA molecules. The **genome** is the complete set of chromosomes in a cell and contains the organism's hereditary information. For example, a human genome has 23 pairs of chromosomes (46 total) with each pair having one chromosome from each parent. A **gene** is a subsequence of a chromosome that contains information for building a protein. Although lengths vary greatly, an average gene contains about 28,000 base pairs [1]. Some contiguous sections of each chromosome are not part of any gene, but some are important for the regulation of gene expression.

8.2 Locations of Genes

We can employ the forward algorithm to solve an important problem in genomics—the locations of genes—by helping us find areas of high CG concentration. In mammalian DNA, the sequence of bases CG appears more frequently upstream of, or before, a gene than in other parts of a DNA sequence, where CG occurs much less than expected from random occurrences of C and G. We call an area of greater concentration of CG a **CpG island**, where “p” in “CpG” represents a phosphate linking the bases C and G. The reason that CG occurs infrequently in much of a sequence of DNA is that often CG transforms to (methyl-C)G before mutating to TG. However, the transformation from CG to TG is suppressed in the CpG islands. A DNA segment of 200 bases is a CpG island if CG occurs at least 50% of the time and the ratio of observed-to-expected number of CpGs is greater than 0.6 [7].

Figure 9a and b contains transition matrices for sequences in and not in CpG islands, respectively. The matrices were derived from a database of human DNA

a		<i>xi</i>				b		<i>xi</i>					
		+	A	C	G	T			+	A	C	G	T
<i>xiI</i>	A	0.180	0.0274	0.426	0.120		A	0.300	0.205	0.285	0.210		
	C	0.171	0.368	0.274	0.188		C	0.322	0.298	0.078	0.302		
	G	0.161	0.339	0.375	0.125		G	0.248	0.246	0.298	0.208		
	T	0.079	0.355	0.384	0.182		T	0.177	0.239	0.292	0.292		

Fig. 9 Possible transition matrix for (a) samples within CpG islands and for (b) samples not within CpG islands [4]

sequences using 48 accepted CpG islands. In the matrix for CpG islands, the probability of the pair CG (or the probability that G occurs, given that C has just appeared) is 0.274, written as $P(x_i = G \mid x_{i-1} = C) = P(G \mid C) = 0.274$, while in the CpG negative matrix (Fig. 9b), the probability of the sequence CG is much lower, $P(G \mid C) = 0.078$. Stanke [21] uses $\pi = (0.148, 0.334, 0.365, 0.154)$ as the initial state probability vector for CpG islands and $\pi = (0.260, 0.249, 0.241, 0.251)$ for non-CpG islands. For each situation, we assume the emission matrix is as follows:

$$\begin{array}{c}
 \textit{hidden/observable} \quad \text{A C G T} \\
 \text{A} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 \text{C} \\
 \text{G} \\
 \text{T}
 \end{array}$$

Project 5 employs these hidden Markov models in accessing the probabilities of output sequences inside and outside CpG islands.

9 Parallel Forward Algorithm (Optional)

Although the forward algorithm has a big improvement in speed over the brute-force technique discussed initially, for a large number of observations and states, such as often encountered in genetics problems, the sequential forward algorithm can take a long time. To speed the task, we can employ parallel programming.

9.1 Communication

One obvious way to parallelize is to have different processes calculate different initial values for the first column and to have different processes compute separate summands, such as those in the middle sections of Figs. 7 and 8. In Holly’s HMM with $s = 2$ states, we employ 2 processes for the initialization step; and in general, for s states, each of s processes could compute a different first-column element. In subsequent steps for Holly’s HMM as in Figs. 7 and 8, each new state is a linear combination of the previous states, so each process calculates the new state by using the inputs from the arrows that point to that state. In theory, s processes are needed for s states to calculate the summands simultaneously. In reality, communication of appropriate forward matrix elements must occur for evaluation of the next column elements, and such communication limits the speedup of the parallel algorithm. Thus, it can often be more efficient to have less processes, so that each process does more work and less communication.

The heart of the forward algorithm is the matrix that holds the probabilities for the sequence of observations. From the perspective of the parallel program, each process could have its own copy of the matrix and would have to communicate the value of its calculated cell to all the other processes. Alternatively, the processes could share the same matrix, which would reduce the amount of needed communication. Such consideration of communication is important in deciding how to implement the parallel algorithm.

9.2 *Implementation of the Parallel Forward Algorithm*

The parallel forward algorithm, available to professors by request, is implemented using the OpenMP library. OpenMP (OMP) uses threads, which are like processes except that the threads share the same memory. Thus, threads communicate by reading and writing to the same matrix.

Communication limits speedup of the algorithm because when multiple threads have to use the same matrix cell, these lightweight processes must take turns—two threads cannot access a cell simultaneously. Thus, increasing the number of threads for any parallel program comes with a tradeoff: Each thread has less work to do, so the work can be done faster; but a thread must wait longer when multiple threads need to access the same matrix cell. Speedup of a parallel program depends on the number of threads used, and increasing the number of threads to its theoretical limit does not always improve speedup.

To illustrate, Table 1 gives timings and speedups and Fig. 10 depicts these timings for serial and parallel OMP implementations of the forward algorithm. One definition of **speedup** using n threads (or processes) is the length of time to execute the serial algorithm divided by the length of time to execute the corresponding parallel algorithm. Notice that the speedup of the parallel algorithm is a function of the number of states and the number of threads. For example, for 2048 states, the speedup is the greatest at about 4 threads:

$$\text{speedup} = 9.9255 \text{ s} / 4.0269 \text{ s} = 2.4648.$$

However, with 4096 states, 8 threads are better:

$$\text{speedup} = 43.6465 \text{ s} / 11.9184 \text{ s} = 3.6621.$$

Even with more states, the use of more threads eventually suppresses speedup because of the need for additional communication.

Table 1 Timings and speedup for forward algorithm

Time in seconds								
# States	Serial	1 thread	2 threads	3 threads	4 threads	8 threads	16 threads	32 threads
512	0.214	0.2490	0.1696	0.1722	0.2473	0.7697	1.3663	2.4829
1024	2.6585	2.7506	1.7230	1.1925	1.0848	1.8645	2.8908	4.9911
2048	9.9255	10.0194	7.2314	5.6453	4.0269	4.1924	5.9881	10.0612
4096	43.6465	44.2792	27.9575	20.2691	16.7255	11.9184	13.4140	20.6784

Speedup								
# States		1 thread	2 threads	3 threads	4 threads	8 threads	16 threads	32 threads
512		0.8593	1.2619	1.2424	0.8653	0.2780	0.1566	0.0862
1024		0.9665	1.5429	2.2293	2.4507	1.4258	0.9196	0.5326
2048		0.9906	1.3726	1.7582	2.4648	2.3675	1.6575	0.9865
4096		0.9857	1.5612	2.1534	2.6096	3.6621	3.2538	2.1107

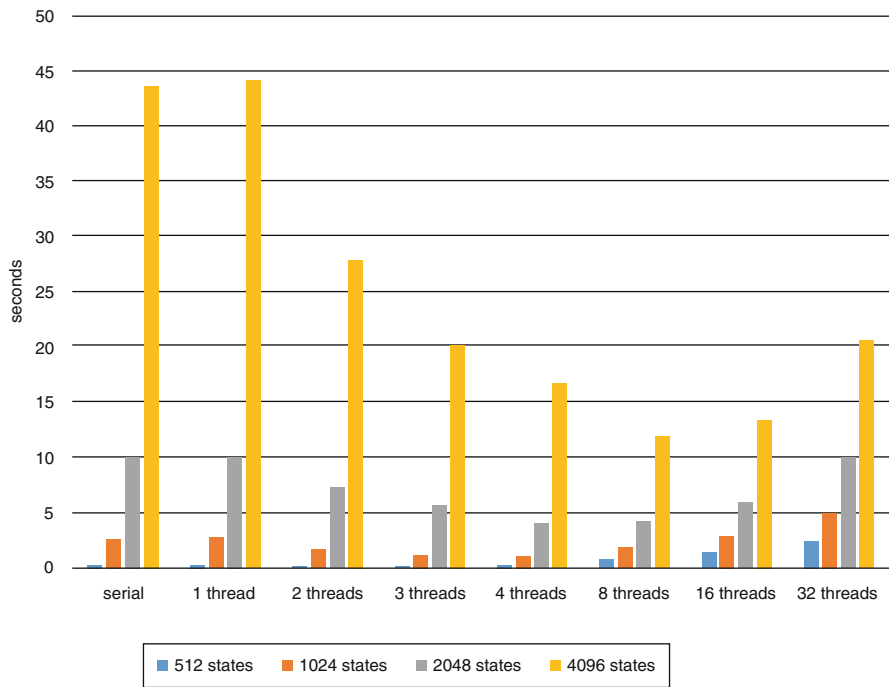


Fig. 10 Timings for forward algorithm

10 Decoding Problem

We employ the HMM forward algorithm to solve likelihood problems, such as the likelihood of monitoring equipment registering breathing sounds (B) and then munching noises (M) the next 2 h for Holly Howler, $P(v = \text{BMM})$. We use the **Viterbi algorithm** to solve another type of HMM problem, **decoding**. In this case, given a sequence of observations, such as $v = \text{BMM}$, we determine the most probable sequence of underlying states, u , to yield v . Thus, we wish to determine the u with maximum $P(u | v)$, such as the sequence of three states, u , that yields $\max(P(u | v = \text{BMM}))$.

10.1 Obvious Solution

In the development above, we learned by joint probability that $P(u, \text{BMM}) = P(u | \text{BMM}) \cdot P(\text{BMM})$. Thus, the problem of determining the state sequence u that maximizes $P(u | \text{BMM})$ is equivalent to the problem of determining the u that maximizes $P(u, \text{BMM})/P(\text{BMM})$. However, because $P(\text{BMM})$ is a constant, the problem simplifies to finding u where $P(u, \text{BMM})$ is maximum. The most obvious method to solve the problem is to determine $P(u, \text{BMM})$ for every possible three-element state sequence, $S^3 = \{\text{RRR, RRE, RER, REE, ERR, ERE, EER, EEE}\}$, using a version of the forward algorithm, and then to select the u with the largest probability. However, this solution is exponentially large because for a sequence of n observations with $h = 2$ states, the number of possible n -element state sequences, or the number of elements in S^n , is $h^n = 2^n$. In general, for h number of states, the number of n -element state sequences is h^n . The much faster **Viterbi algorithm** is another dynamic programming algorithm, which has the forward algorithm as its base.

10.2 Viterbi Algorithm

The key to the Viterbi algorithm is Eq. (14), repeated below, for calculating a joint probability:

$$P(u, v) = P(u_{1,n}, v_{1,n}) = \begin{cases} \pi(u_1) \cdot e(v_1 | u_1), & \text{if } n = 1 \\ P(u_{1,n-1}, v_{1,n-1}) \cdot t(u_{n-1}, u_n) \cdot e(v_n | u_n), & \text{if } n > 1 \end{cases} \quad ((14))$$

Beginning the Viterbi algorithm in the same way as the forward algorithm for observation sequence BMM in Holly's HMM, we employ a 2×3 matrix, G , with first-column elements being $g_{\text{E1}} = P(\text{B}, \text{E}) = \pi(\text{E}) \cdot e(\text{B} | \text{E})$ and $g_{\text{R1}} = P(\text{B}, \text{R}) = \pi(\text{R}) \cdot e(\text{B} | \text{R})$. The initialization step is identical to that of the forward algorithm

(see Fig. 6), resulting in the following initial Viterbi matrix, G :

$$G = \begin{matrix} & & \text{B} & \text{M} & \text{M} \\ \begin{matrix} \text{E} \\ \text{R} \end{matrix} & \begin{bmatrix} 0.06 & \square & \square \\ 0.63 & \square & \square \end{bmatrix} \end{matrix}.$$

Quick Review Question 17

Calculate the first-column elements of the Viterbi matrix to calculate u for $\max(P(u | \text{HHGH}))$ using the HMM in Fig. 11, which contains the following information:

$$S = \{A, B, C\} \text{ and } O = \{G, H\}$$

$$\pi(A) = 0.2, \quad \pi(B) = 0.1, \quad \pi(C) = 0.7$$

$$u_k/u_{k+1} \quad \begin{matrix} \text{A} & \text{B} & \text{C} \end{matrix}$$

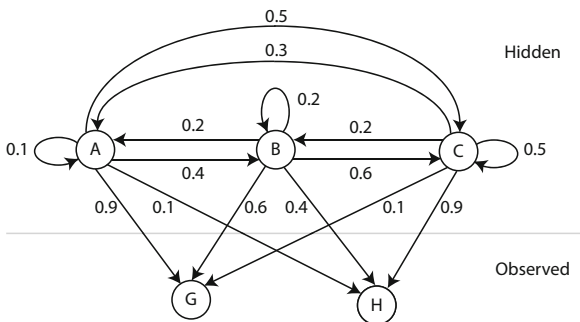
$$T = \begin{matrix} \text{A} \\ \text{B} \\ \text{C} \end{matrix} \begin{bmatrix} 0.1 & 0.4 & 0.5 \\ 0.2 & 0.2 & 0.6 \\ 0.3 & 0.2 & 0.5 \end{bmatrix}$$

$$\text{hidden/observable} \quad \begin{matrix} \text{G} & \text{H} \end{matrix}$$

$$\begin{matrix} \text{A} \\ \text{B} \\ \text{C} \end{matrix} \begin{bmatrix} 0.9 & 0.1 \\ 0.6 & 0.4 \\ 0.1 & 0.9 \end{bmatrix}.$$

Computations of the second column of the Viterbi and the forward matrices also begin in the same way with calculating the product of a first-column element, a transition value, and an emission value. As in Fig. 7 for the forward algorithm, Fig. 12 of the current module for the Viterbi algorithm makes the following

Fig. 11 HMM diagram for Quick Review Question 17



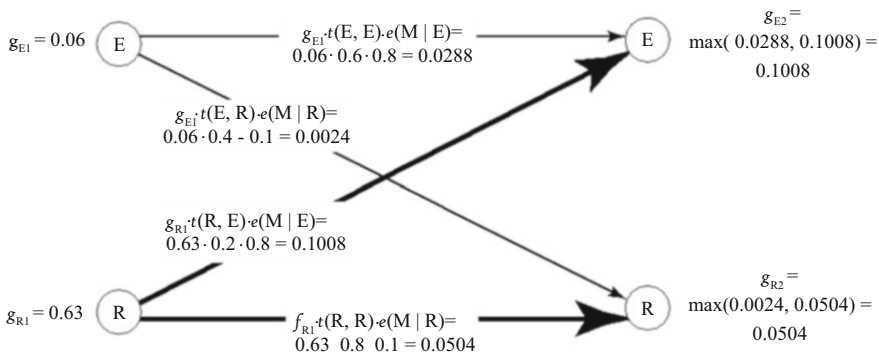


Fig. 12 Calculation of the second column of Viterbi matrix, G

computations using first-column values, g_{E1} and g_{R1} , that correspond to f_{E1} and f_{R1} , respectively, of the forward matrix:

$$\begin{aligned}
 &g_{E1} \cdot t(E, E) \cdot e(M | E) \\
 &g_{E1} \cdot t(E, R) \cdot e(M | R) \\
 &g_{R1} \cdot t(R, E) \cdot e(M | E) \\
 &g_{R1} \cdot t(R, R) \cdot e(M | R)
 \end{aligned}$$

However, instead of taking the sum of pairs of expressions (first and third transitioning to E, second and fourth transitioning to R) as we did with the forward algorithm, we take the maxima as follows:

$$\begin{aligned}
 g_{E2} &= \max(g_{E1} \cdot t(E, E) \cdot e(M | E), \quad g_{R1} \cdot t(R, E) \cdot e(M | E)) \\
 g_{R2} &= \max(g_{E1} \cdot t(E, R) \cdot e(M | R), \quad g_{R1} \cdot t(R, R) \cdot e(M | R)).
 \end{aligned}$$

Figure 12 details these computations with boldface arrows indicating the maxima. The following displays the developing Viterbi matrix, G :

$$G = \begin{matrix} & \begin{matrix} B & M & M \end{matrix} \\ \begin{matrix} E \\ R \end{matrix} & \begin{bmatrix} 0.06 & 0.1008 & \square \\ 0.63 & 0.0504 & \square \end{bmatrix} \end{matrix}$$

Quick Review Question 18

Suppose we wish to use the Viterbi algorithm to find the state sequence, u , with maximum $P(u, HHGH)$ for the HMM in Quick Review Question 17. As calculated in that question, the first column of the Viterbi algorithm matrix G contains $g_{A1} = 0.2$, $g_{B1} = 0.4$, and $g_{C1} = 0.63$.

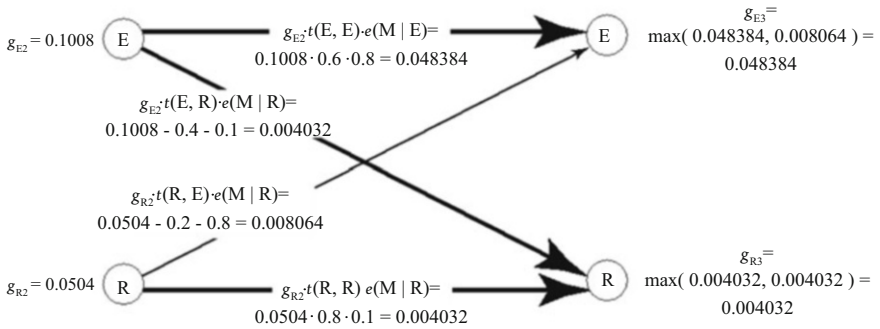


Fig. 13 Step 3 of Viterbi algorithm in calculation of P(BMM)

- a. The calculation of g_{B2} involves three expressions whose values are 0.0032, 0.0032, and 0.0504. Calculate g_{B2} .
- b. Calculate g_{A2} .

Calculations of subsequent Viterbi matrix elements for this example proceed in a similar fashion. With observation v_i , we employ the following evaluations for the elements of column i :

$$\begin{aligned}
 g_{Ei} &= \max(g_{E(i-1)} \cdot t(E, E) \cdot e(v_i | E), & g_{R(i-1)} \cdot t(R, E) \cdot e(v_i | E)) \\
 g_{Ri} &= \max(g_{E(i-1)} \cdot t(E, R) \cdot e(v_i | R), & g_{R(i-1)} \cdot t(R, R) \cdot e(v_i | R))
 \end{aligned}$$

With boldface arrows indicating maximum values, Fig. 13 illustrates the calculation of the final column of the Viterbi matrix. Note that there are two paths to R that yield the maximum, 0.004032. The completed Viterbi matrix is as follows:

$$G = \begin{matrix} & \text{B} & \text{M} & \text{M} \\ \text{E} & \mathbf{0.06} & 0.1008 & 0.048384 \\ \text{R} & \mathbf{0.63} & 0.0504 & 0.004032 \end{matrix}$$

In general, using Viterbi’s algorithm for any HMM; for state, x ; observation, v_i ; and set of states, S , we have the following calculation for a Viterbi matrix element in row x and column i :

$$g_{xi} = \max_{y \in S} (g_{y(i-1)} \cdot (y, x) \cdot e(v_i | x)).$$

Quick Review Question 19

Suppose we wish to use the Viterbi algorithm to find the state sequence, u , with maximum $P(u, HHGH)$ for the HMM in Quick Review Question 17. Suppose, also, that the third column of the Viterbi matrix G contains $g_{A3} = 0.0765$, $g_{B3} = 0.0340$, and $g_{C3} = 0.0142$. Calculate g_{C4} to four decimal places.

To calculate the probability of the visible sequence with the forward algorithm, we added the probabilities in the final column. However, to calculate the maximum joint probability of a hidden sequence and a given visible sequence using the Viterbi algorithm, we find the maximum of the values in the final column. Thus, for Holly’s HMM, we have the following:

$$\max_{u \in S^3}(u, \text{BMM}) = \max(0.048384, 0.004032) = 0.048384.$$

However, we would like to calculate $\max(P(u | \text{BMM}))$ for this u . Recall that

$$P(u, \text{BMM}) = P(u | \text{BMM}) \cdot P(\text{BMM}).$$

Dividing both sides by the factor $P(\text{BMM})$, we have

$$P(u | \text{BMM}) = P(u, \text{BMM})/P(\text{BMM}).$$

Moreover, using the forward algorithm, we discovered $P(\text{BMM}) = 0.080064$. Thus, over all three-element hidden sequences, u ,

$$\max(P(u | \text{BMM})) = 0.048384/0.080064 = 0.604317.$$

More important than finding this maximum probability, we would like to discover the particular state sequence that yields this maximum. Fortunately, by backtracking through the Viterbi matrix, we can determine this hidden sequence. Figure 14 summarizes results of Figs. 9 and 10 with arrows indicating the expressions generating the maxima. To calculate the state sequence, u , that results in $\max(P(u | \text{BMM})) = 0.604317$, we start by finding the maximum in the final column, 0.048384, which is in row E. Backtracking through the path indicated by the arrows, we then go to column 2, row E and finally to column 1, row R. Reading the row values from left to right, we obtain the state sequence $u = \text{REE}$. Thus, given observed sequence BMM, REE is the most likely state sequence, and $P(\text{REE} | \text{BMM}) = 0.604317$, which is over 60%.

Quick Review Question 20

Suppose we wish to use the Viterbi algorithm to find the state sequence, u , with maximum $P(u | \text{HHGH})$ for the HMM in Quick Review Question 17. Suppose, also, that $P(\text{HHGH}) = 0.1028$ and the Viterbi matrix, G , is in Fig. 15, with arrows indicating the direction from which maxima came.

Fig. 14 Final Viterbi matrix with arrows indicating the paths

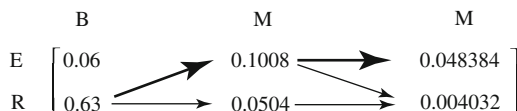
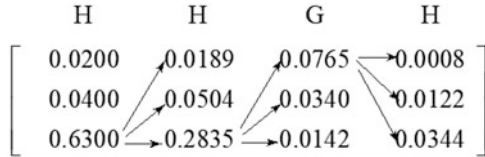


Fig. 15 Viterbi matrix for Quick Review Question 20 with arrows indicating the direction from which maxima came



- a. Calculate the maximum $P(u, \text{HHGH})$ for hidden state sequence u .
- b. Calculate the maximum $P(u | \text{HHGH})$ for hidden state sequence u .
- c. Give the u that achieves these maxima.

10.3 Parallel Viterbi Algorithm (Optional)

As with the forward algorithm, we can use high performance computing to achieve faster results when a decoding problem involves a large number of states and/or observations. Moreover, we can parallelize the Viterbi algorithm similarly to the forward algorithm with OpenMP and threads communicating by reading and writing to the same matrix. Project 6 calculates the speedup that can be achieved using HPC with this algorithm.

11 Detecting CpG Islands

One example of an HMM decoding problem with a solution employing the Viterbi algorithm involves detecting genes. As discussed in section “[Probability of a Genomics Sequence](#),” an area of greater frequency of the base sequence CG can be an indicator that a gene is to follow. The section presented initial probabilities, emission matrices, and possible transition matrices for samples within and not within such CpG islands, called positive and negative areas, respectively [4]. Suppose we also have transition probabilities from bases in positive areas (A^+, C^+, T^+, G^+) to bases in negative areas (A^-, C^-, T^-, G^- , respectively) and vice versa. Then, using the Viterbi algorithm, for a given observed sequence of bases from $\{A, C, T, G\}$, we can compute the most likely hidden sequence from the set of states, $S = \{\text{begin/end}, A^+, C^+, T^+, G^+, A^-, C^-, T^-, G^-\}$, where the sign indicates whether the base is probably in a CpG island or not. Project 7 considers such a decoding problem, where we can decode areas of high CpG concentration, containing bases A^+, C^+, T^+ , and G^+ .

Exercises

- 1 As we did in Quick Review Question 3 using the Venn diagram in Fig. 4, provide an intuitive justification for the first version of Bayes' Theorem in Eq. (6).
- 2 Use a Venn diagram to justify the second version of Bayes' Theorem in Eq. (7).
- 3 This exercise relates to the HMM in Quick Review Question 14.
 - a. Calculate f_{A2} .
 - b. Calculate f_{C2} .
 - c. Calculate $P(HH)$.
 - d. Calculate $P(HHG)$.

Projects

- 1 Write a sequential program to calculate the probability of a state given an observation using Eq. (8).
- 2
 - a. Develop a sequential program to calculate the probability of a sequence of states generating a sequence of observations using Eq. (9).
 - b. Develop a parallel version of this program, having each thread responsible for a portion of the factors and gather the results into a final product.
 - c. For large sequence length, time the parallel version for increasing numbers of threads. Produce a graph of the speedup versus the number of threads similar to Fig. 10.
- 3
 - a. Develop a sequential program to calculate the probability of a state sequence and an observation sequence using Eq. (13).
 - b. Develop a parallel version of this program, having each thread responsible for a portion of the factors and gather the results into a final product.
 - c. For large sequence length, time the parallel version for increasing numbers of threads. Produce a graph of the speedup versus the number of threads similar to Fig. 10.
- 4
 - a. Using the forward algorithm, develop a sequential program to calculate the probability of a sequence of observations.
 - b. Develop a parallel version of this program.
 - c. For large sequence length, time the parallel version for increasing numbers of threads. Produce a graph of the speedup versus the number of threads similar to Fig. 10.
- 5 Download *ProbabilitiesHumanPN.txt* [15], which stores the transition matrices from Fig. 9, and "Accessing Chromosome 19 Data" (*AccessingChr19Data.pdf*) [14], which describes how to access gene locations and subsequences from chromosome 19 of the human genome. Using the UCSC Genome Browser [23], select several subsequences of about 50 bases that occur in a CpG island and several that do not. Using a sequential or parallel forward algorithm program that you develop, determine the probability of each subsequence twice, once for

each hidden Markov model in section “Probability of a Genomics Sequence,” and calculate how many more times likely the subsequence is to be in a CpG island than not. Do your results concur with those of the UCSC Genome Browser?

- 6
 - a. Using the Viterbi algorithm, develop a sequential program to determine the most likely sequence of hidden states corresponding to a sequence of observations.
 - b. Develop a parallel version of this program.
 - c. For a large sequence length, time the parallel version for increasing numbers of threads. Produce a graph of the speedup versus the number of threads similar to Fig. 10.
- 7 Download *ProbabilitiesHumanV.txt* [16], which contains the transition matrix and other data described in the section “Detecting CpG Islands” from [8], and “Accessing Chromosome 19 Data” (*AccessingChr19Data.pdf*) [14], which describes how to access gene locations and subsequences from chromosome 19 of the human genome. Using the UCSC Genome Browser [23], select several subsequences of about 50 bases that overlap CpG islands. Using a sequential or parallel Viterbi algorithm program that you develop, for each downloaded sequence, determine the most likely hidden sequence of states from

$$S = \{\text{begin/end, A}^+, \text{C}^+, \text{T}^+, \text{G}^+, \text{A}^-, \text{C}^-, \text{T}^-, \text{G}^-\}$$

and its probability. Do your results concur with those of the UCSC Genome Browser?

- 8 Scientists have used telemetry data, such as from radio tags or collars, to monitor wildlife movement and HMMs to infer an animal’s hidden behavior [9, 11]. Download *telemetry.nlogo* [17], a NetLogo program to simulate using telemetry to follow an animal and interpret its actions, and “Using a NetLogo Program” (*UsingNetLogo.pdf*) [18], a brief description of how to interact with a NetLogo simulation [12]. With NetLogo, which is free to download, we can generate agent-based models that have autonomous, decision-making agents, which have states and behaviors. In the simulation *telemetry.nlogo*, an animal wanders at random unless thirsty. When thirsty, the animal moves to water, in the center of the world, and stays in the water until satiated. The user can observe, through this simulated telemetry, if the animal is in water (W), facing water but not in water (F), or not facing water and not in water (A). The state of the animal is thirsty, either on way to water or in water (T); or not thirsty, not on way to water and not in water (N).
 - a. Read the description of the model under the Info tab for the program. Estimate the parameters for an HMM to model the scenario as described in the section “Things to Try.”
 - b. Using this HMM, take part of an observation sequence from file output, say of about 50 observations; and with your forward algorithm program, determine the probability of that subsequence.

- c. Using your Viterbi algorithm program and the subsequence of observations from Part b, determine the most likely corresponding state sequence. Calculate the percentage of the sequence derived using the Viterbi algorithm that agrees with the actual state sequence generated by the NetLogo program.
 - d. Revise the NetLogo program, *telemetry.nlogo*, to save to a file at least 100 observation sequences of length 50 and to store in another file the corresponding hidden state sequences. Revise your Viterbi program to read the file of observation sequences and to produce another file of derived state sequences. Write another program to read the latter output file of derived state sequences and the file of state sequences from your NetLogo program and to calculate the fraction of time the two files agree. That is, what is the probability that your Viterbi program will accurately generate the underlying state sequence.
 - e. Extend the model in *telemetry.nlogo* as suggested under the Info tab in the section “Extending the Model.” Tutorials on programming with NetLogo can be obtained from [20] and [12].
- 9 [9] described using telemetry to obtain movement data of bison, which could be in underlying, hidden states of “encamped” or “exploratory.” Their paper discussed several types of HMMs to use observations to infer a bison’s states. Observations were bivariate, involving step length and turning angles, or directions. In an encamped state, the animals were observed to have numerous long steps and few turns; while in an exploratory state, bison were seen having short steps with more frequent reversals.
- a. Similar to *telemetry.nlogo*, available on the website containing this module and discussed in the previous project, develop a NetLogo simulation of bison movement that outputs important totals and writes to one file a sequence of hidden states and to another file the corresponding sequence of observations [17]. In Part b of this project, we will use the output data to estimate an HMM. Then, using this HMM, we will take a sequence of observations and attempt to derive the underlying state sequence. We can compare our derivation with the state sequence from the NetLogo simulation. “Using a NetLogo Program” and tutorials on programming with NetLogo can be obtained from [18] and [20], respectively. Note that we could model the step lengths as being random numbers taken from different exponential distributions, each with step lengths ranging from 0.0 to 0.6 km/d. The turning angles for an exploring and for an encamping animal can be modeled as random numbers from normal distributions with means of 0 and 180 degrees/d, respectively. ([9] employed Weibull distributions for step lengths and wrapped Cauchy distributions for turning angles.)
 - b. Estimate the parameters for an HMM to model the scenario.
 - c. Using this HMM, take part of an observation sequence from file output, say of about 50 observations; and with your forward algorithm program, determine the probability of that subsequence.

- d. Using this HMM, take part of an observation sequence from file output, say of about 50 observations; and with your Viterbi algorithm program, derive the most likely corresponding subsequence of observations. Calculate the percentage of the sequence derived using the Viterbi algorithm that agrees with the actual state sequence generated by the NetLogo program.
 - e. Revise your NetLogo program to save to a file at least 100 observation sequences of length 50 and to store in another file the corresponding hidden state sequences. Revise your Viterbi program to read the file of observation sequences and to produce a file of derived state sequences. Develop another program to take the latter output file of derived state sequences and the file of state sequences from your NetLogo program and to calculate the fraction of time the two files agree. That is, what is the probability that your Viterbi program will accurately generate the underlying state sequence.
- 10 a. In the programming language of your choice (such as Python, R, MATLAB, Mathematica, or C++), develop a program that uses the probabilities from Holly's HMM of this module to generate a given number of states and corresponding observations. Save the sequences to separate files. For example, the initial state should be selected at random using the initial state probabilities. Have the program select the corresponding observation at random with probabilities indicated by the emission matrix. The next state should be picked at random using probabilities from the transition (Markov) matrix, and so forth.
- b. Repeat Parts c–e from Project 9 using this program.
- 11 a. Repeat Project 10 a for a generalized situation. That is, develop a procedure with input of a vector of state names or their abbreviations, a vector of observation names or their abbreviations, an HMM (initial vector, transition matrix, and emission matrix), and a number (n) of states/observations for an output sequence. Then, using the HMM's probabilities, have the procedure generate and return a sequence of n states and corresponding sequence of n observations.
- b. Develop the two HMMs from Project 5. Using your procedure from Part a, generate subsequences of length 50 for the positive and the negative models. Using a forward algorithm program that you develop, determine the probability of each subsequence twice, once for each hidden Markov model. For the positive subsequence, calculate how many more times likely the subsequence is to be in a CpG island than not. For the negative subsequence, calculate how many more times likely the subsequence is not to be in a CpG island than to be in a CpG island.
 - c. A gene in DNA and the corresponding subsequence of RNA have **exons**, which contain part of the encoding information, such as for proteins, separated by **introns**, or non-coding segments. **RNA splicing** removes the introns and reassembles the exons so that translation to the protein can eventually occur, and a **splice** is the location of a cut between an intron and an exon. In DNA and RNA, each non-terminal nucleotide attaches to

the #5 carbon, written **5' carbon**, of a sugar at one end and the #3 carbon, **3' carbon**, of its neighbor at the other end. Thus, we consider a nucleotide chain to have a specific **5'-3' orientation**. Develop the “Toy HMM” for a 5' splice recognition problem in [5], and using your procedure from Part a, repeat Project 9, Parts c–e.

For each of the following HMMs, using your procedure from Part a, repeat Project 9, Parts c–e:

- d. Project 7
 - e. The unfair-casino HMM in Fig. 1 of [3] or other sources
 - f. The weather HMM on the fourth slide of [10]
 - g. The weather HMM in Sections 1 and 2 of [6]
 - h. The light-dark chocolate candy HMM in Section 5, “Exercises,” of [6]
 - i. The tree-growth rings HMM in Section 1, “A Simple Example,” of [13].
- 12 a. Study the Baum-Welch (or forward-backward) algorithm for solving the training (or learning) problem mentioned in sections “[Introduction](#)” and “[Example Model](#),” and develop a program to solve the problem.
- b. Using the NetLogo program *telemetry.nlogo*, available on the website containing this module and discussed in Project 9, generate a collection of state and corresponding observation sequences [17]. Using these and your program from Part a, determine the parameters for an HMM.
- c. With your procedure from Project 10 a, generate a collection of state and corresponding observation sequences. Using these and your program from Part a, determine the parameters for a trained HMM. Compare this HMM to Holly’s HMM.
- Do the same process as described in Part c for the following HMMs:
- d. The two HMMs of Project 11 b
 - e. Project 11 c
 - f. Project 11 d
 - g. Project 11 e
 - h. Project 11 f
 - i. Project 11 g
 - j. Project 11 h
 - k. Project 11 i

Answers to Quick Review Questions

- 1 a. {A, B, C}
- b. {G, H}
- c. 0.6
- d. 0.2
- e.

$$\begin{array}{c}
 u_k/u_{k+1} \quad A \quad B \quad C \\
 T = \begin{array}{c} A \\ B \\ C \end{array} \begin{bmatrix} 0.1 & 0.4 & 0.5 \\ 0.2 & 0.2 & 0.6 \\ 0.3 & 0.2 & 0.5 \end{bmatrix}
 \end{array}$$

- f. 0.2
- g. $0.7 = 1 - \pi(A) - \pi(B)$
- h. 0.6
- i. 0.9
- j.

$$\begin{array}{c}
 \text{hidden/observable} \quad G \quad H \\
 \begin{array}{c} A \\ B \\ C \end{array} \begin{bmatrix} 0.9 & 0.1 \\ 0.6 & 0.4 \\ 0.1 & 0.9 \end{bmatrix}
 \end{array}$$

- 2 a. $P(E, B) = P(B | E) \cdot P(E) = 0.2 \cdot 0.3 = 0.06$
- b. $P(R, M) = P(M | R) \cdot P(R) = 0.1 \cdot 0.7 = 0.07$
- c. $P(R, B) = P(B | R) \cdot P(R) = 0.9 \cdot 0.7 = 0.63$
- d. $1.00 = 0.24 + 0.06 + 0.07 + 0.63$
- e. $0.30 = 0.24 + 0.06$
- f. $0.70 = P(E, M) + P(R, B) = 0.07 + 0.63$
- g. 1.00
- h. $0.31 = P(E, M) + P(R, M) = 0.24 + 0.07$
- i. $0.69 = P(E, B) + P(R, B) = 0.06 + 0.63$
- j. 1.00

3

$$\begin{aligned}
 P(A, B) &\stackrel{?}{=} P(B|A) \cdot P(A) \\
 \frac{c}{a+b+c+d} &\stackrel{?}{=} \frac{c}{\cancel{(a+c)}} \cdot \frac{\cancel{(a+c)}}{a+b+c+d} \\
 \frac{c}{a+b+c+d} &= \frac{c}{a+b+c+d}
 \end{aligned}$$

- 4 a. $0.43 = 0.05 + 0.18 + 0.20$
- b. $0.57 = 0.36 + 0.11 + 0.10$
- c. {F, G, H}
- d. 0.42 because

$$P(G, J) = P(G | J) \cdot P(J), \text{ so } P(G | J) = P(G, J) / P(J) = 0.18 / 0.43 \approx 0.42$$

- e. 0.19 because

$$P(G, K) = P(G | K) \cdot P(K), \text{ so } P(G | K) = P(G, K) / P(K) = 0.11 / 0.57 \approx 0.19$$

f. $\sum_{x \in \{E, G, H\}} P(K, x)$ or $\sum_{x \in \{E, G, H\}} P(x, K)$

g. $0.41 = 0.05 + 0.36$

h. 0.12 because

$$P(J, F) = P(J | F) \cdot P(F), \text{ so } P(J | F) = P(J, F) / P(F) = 0.05 / 0.41 \approx 0.12$$

i. 0.88 because

$$P(K, F) = P(J | F) \cdot P(F), \text{ so } P(K | F) = P(K, F) / P(F) = 0.36 / 0.41 \approx 0.88$$

j. $0.29 = 0.18 + 0.11$

k. $\{J, K\}$

l. $\sum_{x \in \{J, K\}} P(G, x)$ or $\sum_{x \in \{J, K\}} P(x, G)$

m. $0.30 = 0.20 + 0.10$

5 a. 0.62 because $P(J | G) = P(G | J) \cdot P(J) / P(G) = 0.42 \cdot 0.43 / 0.29 \approx 0.62$

b. 0.69 because $P(K | G) = P(G | K) \cdot P(K) / P(G) = 0.35 \cdot 0.57 / 0.29 \approx 0.69$

c. 0.11 because $P(F | J) = P(J | F) \cdot P(F) / P(J) = 0.12 \cdot 0.41 / 0.43 \approx 0.11$

d. 0.63 because $P(F | K) = P(K | F) \cdot P(F) / P(K) = 0.88 \cdot 0.41 / 0.57 \approx 0.63$

6 a. $0.23 \approx 1 - P(E | M) \approx 1 - 0.77$

b. 0.91 because $P(R | B) = P(B | R) \cdot P(R) / (P(B | E) \cdot P(E) + P(B | R) \cdot P(R)) = 0.9 \cdot 0.7 / (0.2 \cdot 0.3 + 0.9 \cdot 0.7) = 0.9 \cdot 0.7 / 0.69 \approx 0.91$

c. $0.09 \approx 1 - P(R | B) \approx 1 - 0.91$; alternatively, $P(E | B) = P(B | E) \cdot P(E) / (P(B | E) \cdot P(E) + P(B | R) \cdot P(R)) = 0.2 \cdot 0.3 / 0.69 \approx 0.09$

7 $0.16 \approx P(F | J) = P(J | F) \cdot P(F) / [P(J | F) \cdot P(F) + P(J | G) \cdot P(G) + P(J | H) \cdot P(H)] = 0.12 \cdot 0.41 / [0.12 \cdot 0.41 + 0.62 \cdot 0.29 + 0.26 \cdot 0.30]$

8 a. $0.0576 = P(\text{MMMB} | \text{ERER}) = e(M | E) \cdot e(M | R) \cdot e(M | E) \cdot e(B | R) = 0.8 \cdot 0.1 \cdot 0.8 \cdot 0.9$

b. $0.53144 \approx P(\text{BBBBBB} | \text{RRRRRR}) = e(B | R)^6 = 0.9^6$

9 a. $0.00055 \approx P(\text{MMMB, ERER}) = P(\text{MMMB} | \text{ERER}) \cdot P(\text{ERER}) = 0.0576 \cdot \pi(E) \cdot t(E, R) \cdot t(R, E) \cdot t(E, R) = 0.0576 \cdot 0.3 \cdot 0.4 \cdot 0.2 \cdot 0.4$, where $P(\text{MMMB} | \text{ERER}) = 0.0576$ is from Quick Review Question 8 a.

b. $0.1219 \approx P(\text{BBBBBB, RRRRRR}) = P(\text{BBBBBB} | \text{RRRRRR}) \cdot P(\text{RRRRRR}) = 0.53144 \cdot \pi(R) \cdot t(R, R)^5 = 0.53144 \cdot 0.70 \cdot 0.8^5$ where $P(\text{BBBBBB} | \text{RRRRRR}) \approx 0.53144$ is from Quick Review Question 8 b.

10 a. $64 = 4^3$

b. $1,048,576 = 4^{10}$

c. $1,099,511,627,776 = 4^{20}$

d. $4,398,046,511,104 = 4^{21}$

11 a. 3×4

b. $0.02 = \pi(H) \cdot e(H | A) = 0.2 \cdot 0.1$

c. $P(A, H)$

- d. $0.04 = \pi(H) \cdot e(H | B) = 0.1 \cdot 0.4$
- e. $P(B, H)$
- f. $0.63 = \pi(H) \cdot e(H | C) = 0.7 \cdot 0.9$
- g. $P(C, H)$
- h. $0.69 = P(A, H) + P(B, H) + P(C, H) = f_{A1} + f_{B1} + f_{C1} = 0.02 + 0.04 + 0.63$
- 12 $0.0504 = P(BM, RR) = P(B, R) \cdot t(R, R) \cdot e(M | R) = f_{R1} \cdot t(R, R) \cdot e(M | R) = 0.63 \cdot 0.8 \cdot 0.1$
- 13 $0.1296 = P(BM, EE) + P(BM, RE) = 0.0288 + 0.1008$
- 14 a. 3
- b. $0.0032 = f_{A1} \cdot t(A, B) \cdot e(H | B) = 0.02 \cdot 0.4 \cdot 0.4$
- c. $0.0032 = f_{B1} \cdot t(B, B) \cdot e(H | B) = 0.04 \cdot 0.2 \cdot 0.4$
- d. $0.0504 = f_{C1} \cdot t(C, B) \cdot e(H | B) = 0.63 \cdot 0.2 \cdot 0.4$
- e. $0.0568 = 0.0032 + 0.0032 + 0.0504$
- 15 a. $P(BMM) = \sum_{x \in S^3} P(BMM, x)$
- b. {EEE, EER, ERE, ERR, REE, RER, RRE, RRR}
- c. $P(BMM) = [P(BMM, EEE) + P(BMM, ERE) + P(BMM, REE) + P(BMM, RRE)] + [P(BMM, EER) + P(BMM, ERR) + P(BMM, RER) + P(BMM, RRR)] = f_{E3} + f_{R3}$
- d. $P(BMM, ERE) + P(BMM, RRE) = P(BM, ER) \cdot t(R, E) \cdot e(M | E) + P(BM, RR) \cdot t(R, E) \cdot e(M | E) = [P(BM, ER) + P(BM, RR)] \cdot t(R, E) \cdot e(M | E) = f_{R2} \cdot t(R, E) \cdot e(M | E)$
- e. $P(BMM, EER) + P(BMM, RER) = f_{E2} \cdot t(E, R) \cdot e(M | R)$
- f. $P(BMM, ERR) + P(BMM, RRR) = f_{R2} \cdot t(R, R) \cdot e(M | R)$
- g. $0.062208 = 0.1296 \cdot 0.6 \cdot 0.8$
- h. $0.005184 = 0.1296 \cdot 0.4 \cdot 0.1$
- i. $0.008448 = 0.0528 \cdot 0.2 \cdot 0.8$
- j. $0.004224 = 0.0528 \cdot 0.8 \cdot 0.1$
- k. $0.70656 = 0.062208 + 0.008448$
- l. $0.009408 = 0.005184 + 0.004224$
- m. $0.080064 = 0.070656 + 0.009408$
- 16 a. 5
- b. 5
- c. h
- d. 5
- e. 2
- f. 2
- g. $95 = 5 \cdot 19$
- h. $h \cdot (n - 1)$
- i. $195 = 5 + 95 \cdot 2$
- j. $2hn - h$ because $h + h \cdot (n - 1) \cdot 2 = h + 2hn - 2h = -h + 2hn$
- k. 0
- l. 4
- m. $h - 1$
- n. $h^2(n - 1) - hn - h$ or $h^2n - h^2 - hn - h$ because $0 + (h - 1) \cdot h \cdot (n - 1) = (h^2 - h)(n - 1) = h^2n - hn - h^2 - h = h^2(n - 1) - hn - h$

- o. $h^2n - h^2 + hn - 2h$ because $(2hn - h) + (h^2n - h^2 - hn - h) = h^2n - h^2 + hn - 2h$
- p. $O(h^2n)$
- 17 $g_{A1} = 0.2, g_{B1} = 0.4,$ and $g_{C1} = 0.63$
- 18 a. $0.0504 = \max(0.0032, 0.0032, 0.0504)$
 b. $0.0189 = \max(0.0002, 0.0008, 0.0189)$
- 19 0.0344 because of the following: $g_{A3} \cdot t(A, C) \cdot e(H|C) = 0.034425$; $g_{B3} \cdot t(B, C) \cdot e(H|C) = 0.01836$; $g_{C3} \cdot t(C, C) \cdot e(H|C) = 0.00639$; and the maximum of these expressions is 0.034425 .
- 20 a. 0.0344 , the maximum in the final column
 b. $0.33463 = 0.0344/0.1028$
 c. CCAC

Further Reading

- Baldi, P., Chauvin, Y., Hunkapiller, T. and McClure, M.A.: Hidden Markov models of biological primary sequence information. Proc. of the Natl. Academy of Sciences, **91**(3), 1059–1063 (1994)
- Gales, M., Young, S.: The application of hidden Markov models in speech recognition. Foundations and Trends in Signal Processing, **1**(3), 195–304 (2007)
- Kamal, M.S., Chowdhury, L., Khan, M.I., Ashour, A.S., Tavares, J.M.R., Dey, N.: Hidden Markov model and Chapman Kolmogrov for protein structures prediction from images. Computational Biology and Chemistry, **68** 231–244 (2017)
- Krogh, A., Brown, M., Mian, I.S., Sjölander, K., Haussler, D.: Hidden Markov models in computational biology: Applications to protein modeling. J. of Molecular Biology, **235**(5), 1501–1531 (1994)
- Manogaran, G., Vijayakumar, V., Varatharajan, R., Kumar, P.M., Sundarasekar, R., Hsu, C.H.: Machine learning based big data processing framework for cancer diagnosis using hidden Markov model and GM clustering. Wireless Personal Communications, **102**(3), 2099–2116 (2018)
- McGibbon, R.T., Ramsundar, B., Sultan, M.M., Kiss, G., Pande, V.S.: Understanding protein dynamics with L1-regularized reversible hidden Markov models. arXiv preprint arXiv:1405.1444 (2014)
- Petersen, B.K., Mayhew, M.B., Ogbuefi, K.O., Liu, V.X., Greene, J.D., Ray, P.: Modeling sepsis disease progression using hidden Markov models (No. LLNL-CONF-740757). Lawrence Livermore National Lab.(LLNL), Livermore, CA (U. S.) (2017)
- Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. of the IEEE **77**(2): 257–286 (1989)
- Ramanathan, N.: Applications of hidden Markov models.
<http://www.cs.umd.edu/~djacobsc/CMSC828/ApplicationsHMMs.pdf>

- Sharp, C., Bray, J., Housden, N.G., Maiden, M.C., Kleanthous, C.: Diversity and distribution of nuclease bacteriocins in bacterial genomes revealed using Hidden Markov Models. *PLoS Computational Biology*, **13**(7), p.e1005652 (2017)
- Williams, J.P., Storlie, C.B., Therneau, T.M., Jack Jr, C.R., Hannig, J.: A Bayesian approach to multi-state hidden Markov models: application to dementia progression. arXiv preprint arXiv:1802.02691 (2018)
- Yoon, B.: Hidden Markov models and their applications in biological sequence analysis. *Curr. Genomics* **10**(6), 402–415 (2009)

Acknowledgements Our thanks go to the Fulbright Specialist Program, University “Magna Graecia” of Catanzaro, and Wofford College for funding the Shiftlets’ visit to the university and to the National Computational Science Institute Blue Waters Student Internship Program for funding Dmitriy Kaplun’s internship.

References

1. B10NUMB3R5: the database of useful biological numbers. <http://bionumbers.hms.harvard.edu/bionumber.aspx?&id=105336&ver=2>
2. Centers for Disease Control and Prevention: 2–20 years: girls stature-for-age and weight-for-age percentiles. <https://www.cdc.gov/growthcharts/data/set2clinical/cj41c072.pdf>
3. Cerulo, L., Ceccarelli, M., Di Penta, M., Canfora, G.: A hidden Markov model to detect coded information islands. In: Source Code Analysis and Manipulation (SCAM) IEEE 13th International Working Conference on Source Code Analysis and Manipulation: 157–166. <https://doi.org/10.1109/SCAM.2013.6648197> (2013)
4. Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G.: *Biological Sequence Analysis*. Cambridge University Press, Cambridge (1998)
5. Eddy, S.R.: What is a hidden Markov model? *Nat Biotechnol.* **22**(10), 1315–6 (2004) <https://doi.org/10.1038/nbt1004-1315>
6. Fosler-Lussier, E.: Markov models and hidden Markov models: a brief tutorial. International Computer Science Inst. <http://di.ubi.pt/~jpaulo/competence/tutorials/hmm-tutorial-1.pdf>
7. Gardiner-Garden, M., Frommer, M. CpG islands in vertebrate genomes. *J. Mol. Biol.* **196**, 261–282 (1987)
8. Huson, D.: Chapter 8: Markov chains and hidden Markov models. In course: Algorithms in Bioinformatics University of Tübingen. <https://ab.inf.uni-tuebingen.de/teaching/ss08/gbi/script/chapter08-hmms.pdf>
9. Langrock, R., King, R., Matthiopoulos, J., Thomas, L., Fortin, D., Morales, J.M.: Flexible and practical modeling of animal telemetry data: hidden Markov models and extensions. *Ecology*, **93**:2336–2342 (2012)
10. Lyngsø, R.: “Hidden Markov models.” http://www.stats.ox.ac.uk/~mcvean/DTC/STAT/Lectures/Weds_wk2/hidden_markov_models.pdf
11. Morales, J.M., Haydon, D.T., Frair, J.L., Holsinger, K.E., Fryxell, J.M.: Extracting more out of relocation data: building movement models as mixtures of random walks. *Ecology* **85**:2436–2445 (2004)
12. NetLogo home page. <https://ccl.northwestern.edu/netlogo/>
13. Stamp, M.: A revealing introduction to hidden Markov models. <https://www.cs.sjsu.edu/~stamp/RUA/HMM.pdf>
14. Shiftlet, A.: Accessing Chromosome 19 data. AccessingChr19Data.pdf in <https://ics.wofford-ecs.org/files/HMM.zip>

15. Shiflet, A.: Possible transition matrices for samples within and not within CpG islands. ProbabilitiesHumanPN.txt in <https://ics.wofford-ecs.org/files/HMM.zip>
16. Shiflet, A.: Transition matrix and other data from Hudson. ProbabilitiesHumanV.txt in <https://ics.wofford-ecs.org/files/HMM.zip>
17. Shiflet, A.: Telemetry program telemetry.nlogo in NetLogo. telemetry.nlogo in <https://wofford-ecs.org/files/HMM.zip>
18. Shiflet, A. Using a NetLogo program. UsingNetLogo.pdf in <https://ics.wofford-ecs.org/files/HMM.zip>
19. Shiflet, A., Shiflet, G. Introduction to Computational Science: Modeling and Simulation for the Sciences, 2nd ed., Princeton University Press (2014)
20. Shiflet, A., Shiflet, G. NetLogo agent-based files. <https://ics.wofford-ecs.org/agent/NetLogo>
21. Stanke, M.: Markov chains and hidden Markov models, Free University of Berlin. <http://www.mi.fu-berlin.de/wiki/pub/ABI/HiddenMarkovModelsWS13/script.pdf>
22. Sungkaworn, T., Jobin, M.L., Burnecki, K., Weron, A., Lohse, M.J., Calebiro, D.: Single-molecule imaging reveals receptor-G protein interactions at cell surface hot spots. *Nature* **550**(7677) 543–547 (2017)
23. UCSC Genome Browser. <https://genome.ucsc.edu/cgi-bin/hgGateway>