

# MS2a, Week 5, Model Solution

Rune Lyngsø

November 15, 2011

## A Hidden Markov Model Use

- a. Consider a hidden Markov model emitting sequences over the alphabet  $\{A, C, G, T\}$ . The model has two states, 0 and 1, that are equiprobable start states. Transition probabilities are 0.75 for remaining in a state and 0.25 for switching to the other state. In state 0 A and G are emitted with probability 0.40 while C and T are emitted with probability 0.10. In state 1 A and G are emitted with probability 0.10 while C and T are emitted with probability 0.40. What is the probability of observing the sequence ACTG? Observe that we do not have an end state providing explicit termination, so the model will not model a sequence length distribution. Rather, for every sequence length it models a distribution over sequence content.

The probability of the observation is the sum of the probabilities of all possible ways the hidden Markov model can generate ACTG. Hence we need to invoke the forward algorithm computing the sum of probabilities of paths emitting ACTG, and finally summing over all possible states we could end in rather than just look up the probability for ending in the end state. The table computed by the forward algorithm is

1	0.05	0.035	0.012125	0.000962
0	0.2	0.01625	0.002094	0.001841
	A	C	T	G

The sum of probabilities in the last column is  $2.80 \cdot 10^{-3}$ .

- b. What is the most likely sequence of hidden states, and how probable is it?

Here we need to find the most likely path in the HMM generating ACTG, rather than the sum over all paths, and hence it is the Viterbi

algorithm that should be applied. Doing this, we get the following table with backtrack of the largest value in the last column shown by arrows:

1	0.05	0.02	0.0086	0.00045
0	0.2	0.015	0.001125	0.0006
	A	C	T	G

So the most likely annotation of the sequence with states is  ${}^0 1 1 0$  **ACGT** which has probability  $9.61 \cdot 10^{-4}$ . Note that each symbol is annotated with the state most likely to emit it. This is not surprising if we take a closer look at the probabilities: the ratio between emission probabilities is a third larger than the ratio between the transition probabilities. Still, the most likely path contributes less than a quarter of the total probability of the sequence.

- c. What is the most likely hidden state at position 2, summing over all possible paths, and how probable is it?

We have just determined the state at position 2 in the most likely sequence of hidden states. But here we need to sum over all possible sequences of hidden states. Hence we need to combine the probability of all paths generating the sequence up to position 2 with all paths generating the sequence after position 2. I.e. we need to invoke the backward algorithm. The table computed by the backward algorithm, where we for convenience omit the emission probability of the first symbol, is

1	0.019234	0.060625	0.175	1
0	0.009203	0.041875	0.325	1
	A	C	T	G

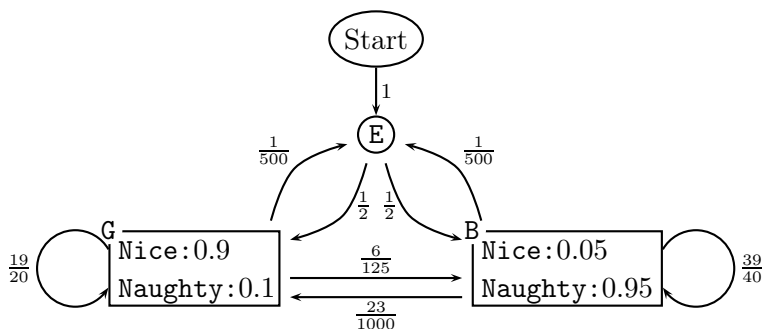
Multiplying with the corresponding entries of the forward table we get a contribution to the total probability from paths emitting the C from state 0 of  $6.80 \cdot 10^{-4}$  and a contribution from state 1 of  $2.12 \cdot 10^{-3}$  (observe that these two probabilities sum to the total probability of ACTG, as they should). Hence state 1 is the most probable at position 2, with a posterior probability of 76%.

## B Hidden Markov Model Design

- d. The occasionally dishonest casino is a standard HMM example. Given the season and the problems ludomania causes in modern society, we will consider a rephrased version of this example. **Infinity Road** is an endless row of houses with a child living in most. Given the length of the street, Santa and his elves cannot constantly check the behaviour of all the children in the road, but checks just one deed per child each year to see whether it is **Naughty** or **Nice**. However, even a **Nice** child may transgress and perform a **Naughty** act (with probability 10% i.i.d. for all **Nice** children), and a **Naughty** child may inadvertently find himself doing something that can be classified as **Nice** (with probability 5%, again i.i.d. for all **Naughty** children). Santa would like to do better than basing his judgement on just a single observation, and it just so happens that **Infinity Road** segments into **Good** neighbourhoods and **Bad** neighbourhoods, both of lengths that are geometrically distributed and with an expected length of a **Good** neighbourhood of 20 houses and an expected length of a **Bad** neighbourhood of 40 houses. All children living in a **Good** neighbourhood are **Nice**, and all children living in a **Bad** neighbourhood are **Naughty** (it's all about peer pressure). It is equally likely that **Infinity Road** starts with a **Good** neighbourhood as with a **Bad** neighbourhood. Houses with no children living in them are distributed uniformly at random, with on average one out of every 500 houses not having a child living in it. Unfortunately the records of which houses are childless has been lost, all Santa has to go by is the sequence of **Nice** and **Naughty** for the deed checked for each child as you go down **Infinity Road**. Neighbourhoods either side of one or more childless houses are uncorrelated, such that the neighbourhood starting after a childless house has equal chance of being **Good** and **Bad**. Design a hidden Markov model that can help Santa use the observations for all the children to annotate each child as either **Naughty** or **Nice** – don't worry that it would normally take infinitely long time to annotate an infinitely long sequence.

The annotation problem is one of annotating each deed, whether **Naughty** or **Nice**, with a neighbourhood state, either **Good** or **Bad**, as the neighbourhood state yields the overall inclination of a child regardless of the character of the observed deed. If we start with the childless houses, there are no observations for these so a *silent* state seems appropriate. From all states, we need to go to this silent state with probability  $\frac{1}{500}$ . This also holds for the state itself, but as it is silent we may as well skip the self-loop on this state and just transition to either a **Good**

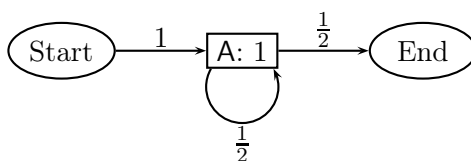
or a **Bad** neighbourhood with equal probability  $\frac{1}{2}$ . This means that we may also use this for the initial distribution. Otherwise we need two *non-silent* states, one for each type of neighbourhood, and the size expectations immediately specifies the probability of the self-loop transition for each. The remaining probability is assigned the transition switching to the other type of neighbourhood. These observations give the following HMM



where the state **G** models houses in Good neighbourhoods, the state **B** models houses in Bad neighbourhoods, and the state **E** models childless houses.

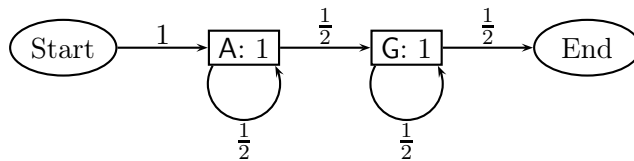
- e. Construct a HMM that generates the sequence  $A^i$ , i.e. the sequence of  $i$  As, with probability  $2^{-i}$  for  $i \geq 1$ , if possible. Otherwise argue it is not possible.

This is just a geometric distribution on  $i$  with parameter  $1/2$ , so we just need a single state emitting the A's, with probability  $1/2$  of staying in the state after an emission and probability  $1/2$  of going to the end state:



- f. Construct a HMM that generates sequences over the alphabet  $\{A, G\}$  with probability  $(k-1)2^{-k}$  for generating a sequence of length  $k \geq 2$ , and for which all sequences that are generated of length  $k$  are on the form  $A^i G^{k-i}$  with  $1 \leq i < k$  and equiprobable.

Consider the HMM



Evidently it can only generate sequences on the form  $A^i G^{k-i}$ ,  $k \geq 2, 1 \leq i < k$ , and for a given  $k$  all sequences that can be generated of length  $k$  have probability  $2^{-k}$ : there is a unique run generating a particular sequence of length  $k$ , and in this run all emissions have probability 1 while there is one transition with probability 1 and  $k$  transitions with probability  $\frac{1}{2}$ . It follows that the total probability of generating a sequence of length  $k$  is  $(k-1)2^{-k}$ .

- g. Construct a HMM that generates the sequence  $A^i G^i$  with probability  $2^{-i}$  for  $i \geq 1$ , if possible. Otherwise argue that it is not possible.

For this to work, we would need to know how many A's were emitted when we start emitting the G's. But this means that we need to be able to remember arbitrarily far back, which should tell us we will be violating the Markov property.

More formally, assume that there is a hidden Markov model  $M$  with  $n$  states that emits sequences according to the required distribution. Consider a run emitting the sequence  $A^n G^n$ . This will pass through  $2n$  states, so there must be a state  $q$  that is visited at least twice. We can write  $A^n G^n$  as  $xyz$ , where the non-empty sequence  $y$  is the part of  $A^n G^n$  emitted between the first visit to  $q$  and the last visit to  $q$ . Evidently the sequence  $xyyz$  can be generated with non-zero probability, as we could repeat the path taken between the first visit to  $q$  and the last visit to  $q$  one more time. As  $M$  cannot emit sequences with G's preceding A's,  $y$  must be on the form  $A^i G^j$ . But if  $i \neq j$ , then  $xyyz$  does not contain the same number of A's and G's and should thus be emitted with probability zero. Hence  $i = j > 0$ , as  $y$  is non-empty. But then  $xyyz = A^n G^i A^i G^n$ , which conflicts the fact that G's cannot precede A's. Consequently our assumption that there was a (finite) hidden Markov model generating sequences according to the specified distribution must have been wrong.

### C RNA Secondary Structure Prediction

- h. Use Algorithm 1 and Algorithm 2 of the lecture notes on RNA secondary structure prediction to find the maximum number of base pairs for the

sequence GAGGCU, and a structure with this number of base pairs. Two bases can form a valid base pair if *i*) they are separated by at least three bases in the sequence, i.e. their indices differ by at least 4, and *ii*) they form one of the three types of base pairs shown in Figure 2 in the lecture notes. For added convenience, the table you need to fill out and backtrack (cf. Figure 5 in the lecture notes) is:

		second base # →							
	0	G <sub>1</sub>	A <sub>2</sub>	G <sub>3</sub>	G <sub>4</sub>	C <sub>5</sub>	U <sub>6</sub>		
first base # ↓	G <sub>1</sub>	0	0	0	0	0	1	1	G <sub>1</sub>
	A <sub>2</sub>		0	0	0	0	0	1	A <sub>2</sub>
	G <sub>3</sub>			0	0	0	0	0	G <sub>3</sub>
	G <sub>4</sub>				0	0	0	0	G <sub>4</sub>
	C <sub>5</sub>					0	0	0	C <sub>5</sub>
	U <sub>6</sub>						0	0	U <sub>6</sub>
	7							0	7

The maximum number of valid base pairs that can be formed, 1, is the entry in the upper righthand corner of the triangular table. There are two different ways to backtrack this number, indicated by dashed and solid arrows respectively. One corresponds to the secondary structure consisting of the base pair C<sub>1</sub> · G<sub>5</sub>, and the other corresponds to the secondary structure consisting of the base pair A<sub>2</sub> · U<sub>6</sub>.

- i. Forgetting about Algorithms 1 and 2, can you find a structure with more valid base pairs than the one you found above? If so, why does Algorithm 1 fail to find this number of base pairs?

Both G<sub>1</sub> · C<sub>5</sub> and A<sub>2</sub> · U<sub>6</sub> are valid base pairs, and they do not share any bases. So we can form a structure with these 2 base pairs. The reason that Algorithms 1 and 2 don't find this structure is that the two base pairs are crossing. Algorithm 1 only considers structures without crossing base pairs, also known as pseudoknots.

- j. How many different structures with no crossing base pairs can you find for this sequence?

There are three pairs of bases with the minimum required separation,  $G_1 \cdot C_5$ ,  $A_2 \cdot U_6$ , and  $G_1 \cdot U_6$ , are canonical base pairs. However, any two of these pairs will either share a base or cross each other, so the possible structures are the three structures containing exactly one of these base pairs and the structure containing no base pairs.

- k. Describe a recursion for determining the number of different possible structures possible for an RNA sequence  $s$ . Equation (1) in the RNA notes is a perfect source for inspiration, but remember that we need to count all structures rather than finding the score of the best one.

Equation (1) finds the best score taken over all structures, but here we need to sum the ‘scores’ of all structures where the ‘score’ of each structure is 1 (it contributes 1 to the total count). Hence, we need to add values instead of taking the maximum, and when we have contributions from separate parts of the sequence this should be multiplied rather than added – if we have  $m$  choices for one part and  $n$  choices for the other part and these are independent, we have a total of  $mn$  choices for the combination. This results in the following recursion for the number of structures possible for a sequence  $s$ .

$$N(i, j) = \begin{cases} 1 & \text{if } j \leq i + 3 \\ N(i + 1, j) + \sum_{\substack{i+3 < k \leq j \\ s[i] = s[k]}} N(i + 1, k - 1) \cdot N(k + 1, j) & \text{otherwise} \end{cases} \quad (1)$$

Would it be as easy to modify equations (2) and (3) to obtain a recursion for the number of structures when all base pairs are required to be stacking, i.e. have a neighbouring base pair (why/why not)?

It wouldn’t be quite as easy to modify equations (2) and (3) to count structures when base pairs are required to be stacking. If we just applied the same modifications as for equation (1) we would overcount the number of structures. The cases that we split the recursion into in equation (1) are disjoint, as each case has the first base either unpaired or base paired to a specific base. In the recursion for  $V(i, j)$  in equation (2) we have the first and the last base form a base pair in both cases. As  $W(i + 1, j - 1)$  does not exclude the case where bases  $i + 1$  and  $j - 1$  are paired, this would result in the some structures contributing through both choices.