

# Hidden Markov Models

(Algorithms: forward, backward and Baum-Welch)

Tuesday 13<sup>th</sup> January 2015

Dr Daniel Wilson

Dr Chieh-Hsi (Jessie) Wu

Nuffield Department of Medicine

# Probability of an observed sequence

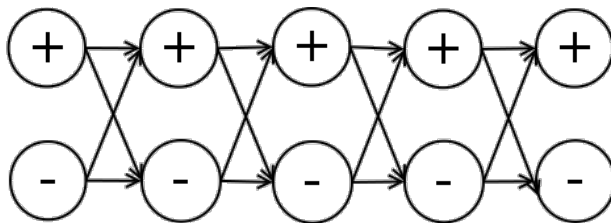
- One might be interested to find the full probability of a sequence of observations,  $x$ , given the HMM model,  $P(x | \theta)$ , where  $\theta$  represents the HMM model parameters.
  - This is also called the likelihood.
- Calculating  $P(x | \theta)$  requires adding up the probabilities for all possible hidden state paths in the HMM

$$P(x | \theta) = \sum_h P(x, h | \theta)$$

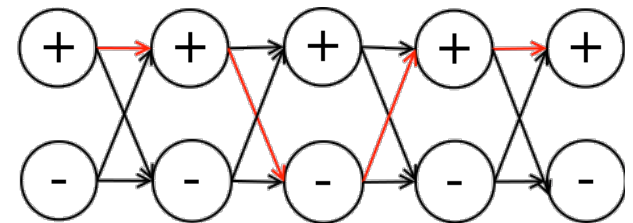
where  $h$  is the path of hidden states.

# Calculating $P(x | \theta)$ - Example

- Consider the gene-finding example:
  - Suppose we have a sequence 5 nucleotide triplets (S, N, S, S, N), where S codes amino acids and N codes for the stop codons.
  - Each of these triplets might be in a gene (+) or in the intergenic (-) region.



Observations   S       S       N       S       N



Observations   S       S       N       S       N

# Calculating $P(x | \theta)$ – Brute Force

- For this example, there would be  $2^5 = 32$  possible hidden state paths and the full probability is given by

$$P(S, S, N, S, N | \theta) =$$

$$P(S, S, N, S, N, +, +, +, +, + | \theta) +$$

$$P(S, S, N, S, N, +, +, +, +, - | \theta) +$$

$$P(S, S, N, S, N, +, +, +, -, + | \theta) +$$

... +

$$P(S, S, N, S, N, -, -, -, -, - | \theta)$$

- It is computationally expensive to calculate the probability by enumerating all possible paths.

# Calculating $P(x | \theta)$ – Brute Force

- The number of possible paths increases exponentially with the length sequence.
- If there are  $L$  observations and  $K$  hidden states, then there are  $K^L$  possible hidden state paths.
- In reality, a nucleotide sequence of interest could have millions of triplets, so the number of possible hidden state paths is astronomical even when there are only two hidden states.
- The full probability can be calculated by a dynamic programming procedure similar to the Viterbi algorithm.

# Calculating $P(x | \theta)$ – Forward algorithm

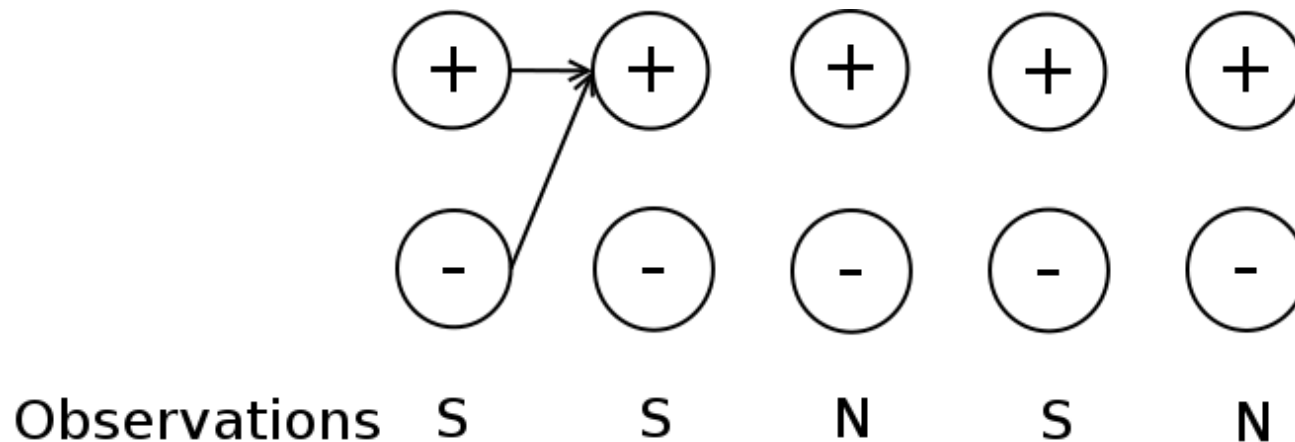
- The algorithm computes partial probabilities
  - A partial probability is the probability of reaching an intermediate state in the diagram

$$P(x_1, \dots, x_i, h_i = k | \theta)$$

- These partial probabilities are computed at all sequence positions.

# Calculating $P(x | \theta)$ – Forward algorithm

- We calculate the probability of reaching an intermediate hidden state as the sum of all the possible paths to that state.
- For example, the probability that the second triplet is in a gene,  $P(x_1 = S, x_2 = S, h_2 = + | \theta)$ , is calculated from all the paths represented by the arrows in the diagram



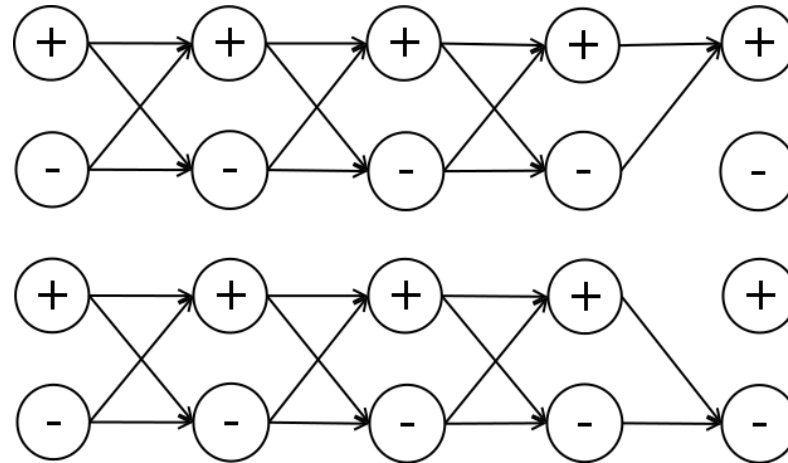
# Forward algorithm – Partial Probabilities

- Let the term  $f_k(i) = P(x_1, \dots, x_i, h_i = k | \theta)$ .
- $f_k(i) = P(\text{observation at } i | \text{hidden state is } k) \times P(\text{all paths to state } k \text{ at position } i)$ .
- The partial probabilities for observation at the last position ( $L$ ) represents the probability of reaching those states going through all possible paths.



# Forward algorithm

- For the gene-finding example, the final probabilities are calculated from the paths shown below:



- Summing up these final partial probabilities gives the sum of all possible paths of the hidden states – the probability of observing the sequence given the HMM.

# Forward algorithm

- Algorithm:
  - Initialisation ( $i = 1$ ):

$$f_1(k) = \pi_k e_k(x_1)$$

- Recursion ( $i = 2, \dots, L$ ):

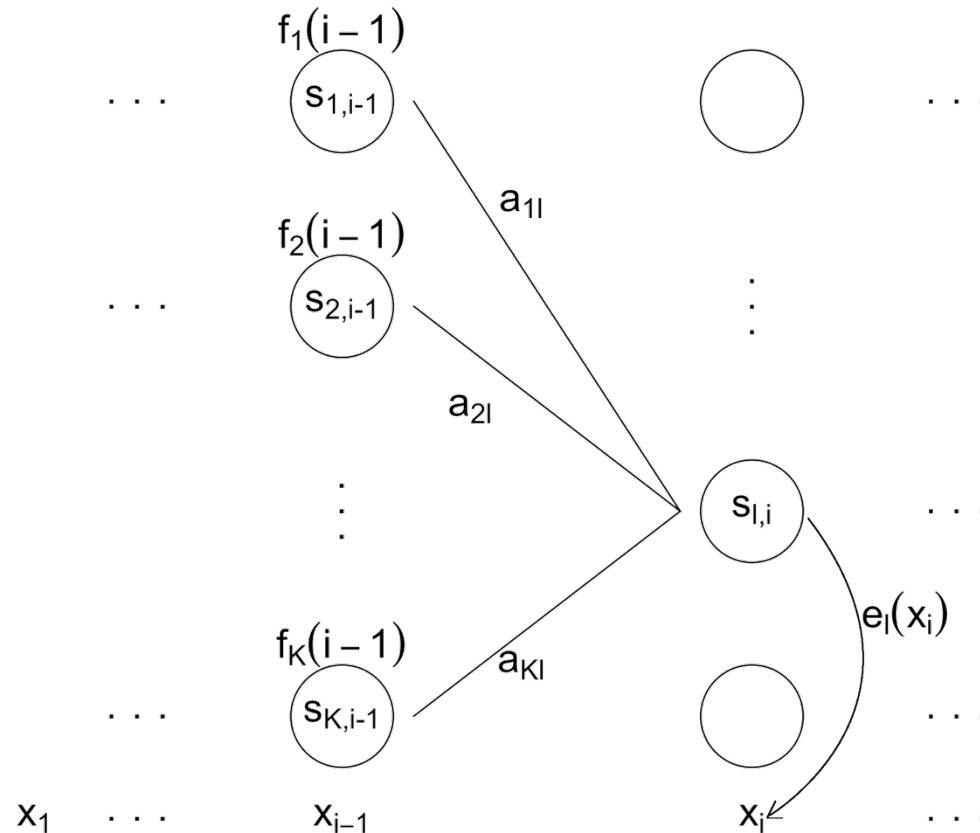
$$f_i(i) = e_i(x_i) \sum_k f_k(i-1) a_{ki}$$

- Termination

$$P(x|\theta) = \sum_k f_k(L)$$

# Forward algorithm - Visualisation

$$f_l(i) = P(x_1, \dots, x_i, h_i = l \mid \theta)$$



$$f_l(i) = e_l(x_i) \sum_k f_k(i-1) a_{kl}$$

# Backward algorithm

- Sometimes, we would like to know what is the probability that hidden state  $j$  produced the observation at position  $i$ ,  $x_i$ ,  $P(h_i = j | x, \theta)$ .
- This requires first calculating the joint probability  $P(x, h_i = j | \theta)$  and then  $P(h_i = j | x, \theta)$  can be calculated using  $P(x, h_i = j | \theta) / P(x | \theta)$ .

# Backward algorithm

- The required joint probability can be re-written as

$$\begin{aligned} P(x, h_i = k \mid \theta) \\ &= P(x_1, \dots, x_i, h_i = k \mid \theta) P(x_{i+1}, \dots, x_L \mid x_1, \dots, x_i, h_i = k, \theta) \\ &= P(x_1, \dots, x_i, h_i = k \mid \theta) P(x_{i+1}, \dots, x_L \mid h_i = k, \theta) \end{aligned}$$

- The first term,  $P(x_1, \dots, x_i, \pi_i = k \mid \theta)$ , is computed using the forward algorithm.
- The second term is obtained by a backward recursion starting at the end of the sequence

# Backward algorithm

- Let  $b_k(i) = P(x_{i+1}, \dots, x_L | h_i = k, \theta)$ , which represents the probability that the partial observation  $x_{i+1}, \dots, x_L$  is generated given the hidden state at position  $i$  is  $k$ .
- The  $b_k(i)$  is analogous to the forward probability and can be obtained by the backward algorithm.

# Backward algorithm

- Algorithm:

- Initialisation ( $i = L$ ):

$$b_k(L) = 1 \text{ for all } k$$

- Recursion ( $i = L - 1, \dots, 1$ ):

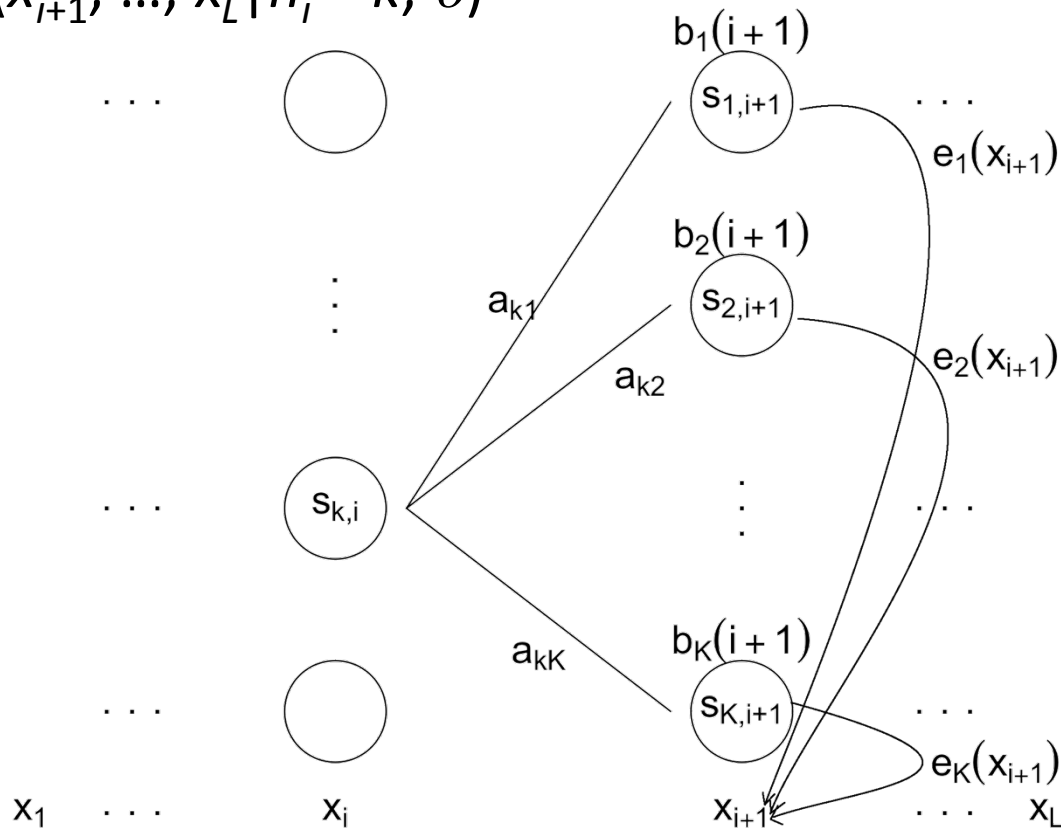
$$b_k(i) = \sum_l a_{kl} e_l(x_{i+1}) b_l(i+1)$$

- Termination

$$P(x | \theta) = \sum_k \pi_k e_k(x_1) b_k(1)$$

# Backward algorithm - Visualisation

$$b_k(i) = P(x_{i+1}, \dots, x_L | h_i = k, \theta)$$



$$b_k(i) = \sum_l a_{kl} e_l(x_{i+1}) b_l(i+1)$$



# Parameter estimation

- In the examples presented so far, we have been given *a priori* the values of the parameters of the HMM, including
  - the hidden state transition matrix,  $\mathbf{a}$
  - the initial distribution of the hidden state,  $\pi$
  - the emission probabilities,  $\mathbf{e}$ .
- However, in practice, these parameters are often not known and therefore have to be estimated from the data.
- Let  $\theta = (\mathbf{a}, \pi, \mathbf{e})$ . The goal is to find parameter values that maximizes the likelihood (the full probability of a sequence of observations),  $P(x | \theta)$ , in other words,

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} P(x | \theta)$$

# Parameter estimation for HMMs

- When the hidden state path is known
  - The following can be calculated:
    - $A_{kl}$  = # times transition from hidden state  $k$  to state  $l$  occurred
    - $E_k(j)$  = #times hidden state  $k$  produces observation  $j$ .
  - The maximum likelihood estimator for  $a_{kl}$  and  $e_k(j)$  are

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}} \quad \text{and} \quad e_k(j) = \frac{E_k(j)}{\sum_{j'} E_k(j')}$$

# Parameter estimation for HMMs

- When the hidden state sequence is UNknown
  - The maximum likelihood estimate of  $\theta$  cannot be calculated analytically.
  - Algorithms for optimisation have to be used.
  - Baum-Welch algorithm
    - Expectation-Maximization (EMs) algorithm
    - The algorithm starts with some initial value, then iteratively re-estimates the parameters to improve  $P(x|$  new values of  $\theta)$ .

# Baum-Welch Algorithm

- Re-estimation of the transition probabilities
  - Given the current model parameters and the sequence of observations, the probability of being at hidden state  $k$  at position  $i$  going to state  $l$  at position  $i + 1$  is

$$\begin{aligned}\xi_i(k, l) &= P(h_i = k, h_{i+1} = l \mid x, \theta) \\ &= \frac{f_k(i) a_{kl} e_l(x_{i+1}) b_l(i+1)}{\sum_{k=1}^K \sum_{l=1}^K f_k(i) a_{kl} e_l(x_{i+1}) b_l(i+1)}\end{aligned}$$

# Baum-Welch Algorithm

- Re-estimation of the transition probabilities
  - Intuitively

$$\hat{a}_{kl} = \frac{\text{expected number of transitions from state } k \text{ to state } l}{\text{expected number of transitions from state } k}$$

- Formally

$$\hat{a}_{kl} = \frac{\sum_{i=1}^{L-1} \xi_i(k, l)}{\sum_{i=1}^{L-1} \sum_{l'=1}^K \xi_i(k, l')}$$

# Baum-Welch Algorithm

- Re-estimation of the transition probabilities
  - We define the state probability at position  $i$  as

$$\gamma_i(k) = \sum_{l=1}^K \xi_i(k, l)$$

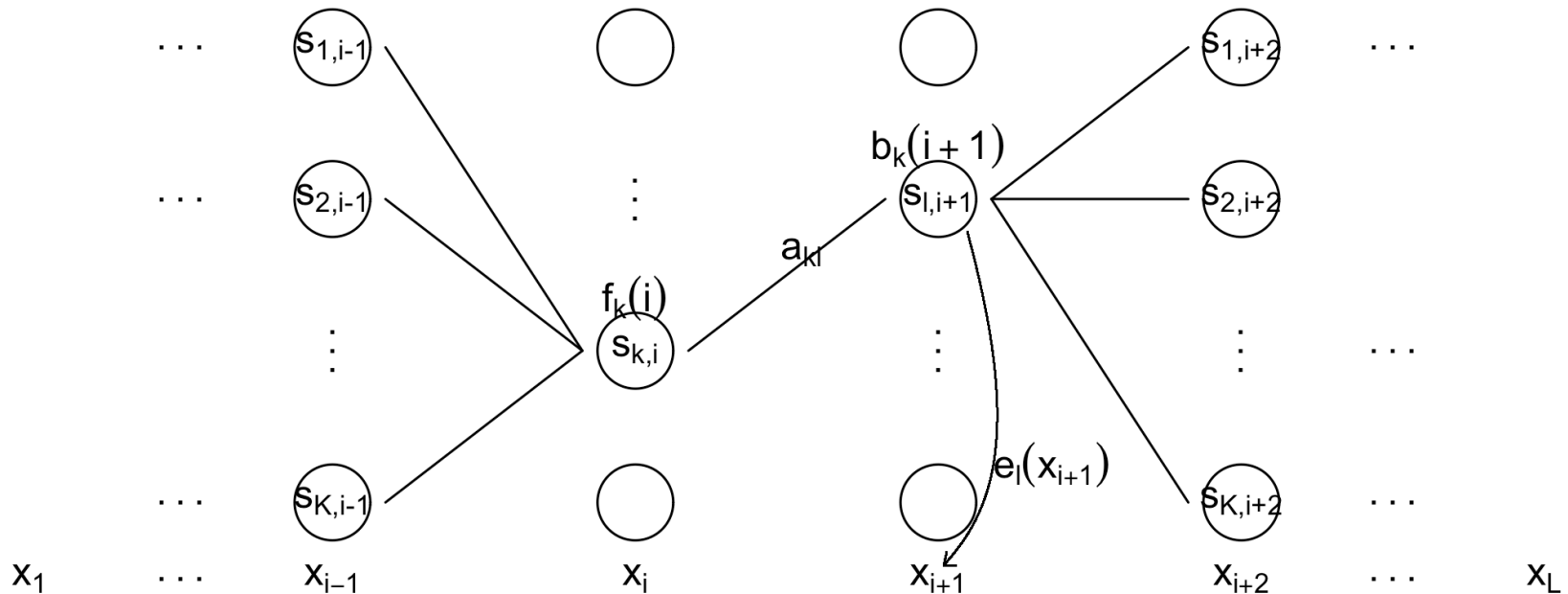
which is the probability of being in state  $k$  at position  $i$  given the complete sequence of observations,  $x$ .

- The re-estimated transition probabilities can be defined as

$$\hat{a}_{kl} = \frac{\sum_{i=1}^{L-1} \xi_i(k, l)}{\sum_{i=1}^{L-1} \gamma_i(k)}$$

# Baum-Welch Algorithm - Visualisation

$$\xi_i(k,l) = P(h_i = k, h_{i+1} = l | x, \theta)$$



$$\xi_i(k,l) = \frac{f_k(i) a_{kl} e_l(x_{i+1}) b_l(i+1)}{\sum_{k=1}^K \sum_{l=1}^K f_k(i) a_{kl} e_l(x_{i+1}) b_l(i+1)}$$

# Baum-Welch Algorithm

- Re-estimation of the initial hidden state distribution
  - Re-estimation of the probability that state  $k$  is the hidden start state
  - Intuitively

$\hat{\pi}_k$  = expected number of times in state  $k$  at  $i = 1$

- Formally

$$\hat{\pi}_k = \gamma_1(k)$$



# Baum-Welch Algorithm

- Re-estimation of the emission probabilities
  - Intuitively

$$\hat{e}_k(j) = \frac{\text{expected number of times in state } k \text{ and observed outcome } j}{\text{expected number of times in state } k}$$

- Formally

$$\hat{e}_k(j) = \frac{\sum_{i=1}^{L-1} \delta(x_i, j) \gamma_i(k)}{\sum_{i=1}^L \gamma_i(k)}$$

where  $\delta(x_i, j) = 1$ , if  $x_i = j$ , and 0 otherwise.

# Baum-Welch Algorithm

- Initialisation
  - Pick arbitrary model parameters
- Recurrence
  - Calculate  $f_k(i)$  using the forward algorithm
  - Calculate  $b_k(i)$  using the backward algorithm
  - Using the rules presented earlier, update  $\hat{a}_{kl}$ ,  $\hat{e}_k(j)$  and  $\hat{\pi}_k$
  - Calculate the new log likelihood of the model
- Termination
  - Stop if the change in log likelihood is less than some predefined threshold or the maximum number of iterations is exceeded.

# Practical

- Implement the forward algorithm to calculate the full probability of the sequence you have downloaded in the previous practical session.
  - Compare this probability to the probability given the hidden state path found by the Viterbi algorithm.
- Implement the backward algorithm for the same sequence.
  - Compare the probability calculated here to that calculated using the forward algorithm.
  - Calculate the probabilities of being in a gene and the intergenic region at each position given the sequence.
- Bonus: Implement the Baum-Welch algorithm to estimate the parameters of HMM