

# Hidden Markov Models II

Machine Learning 10-601B

Seyoung Kim

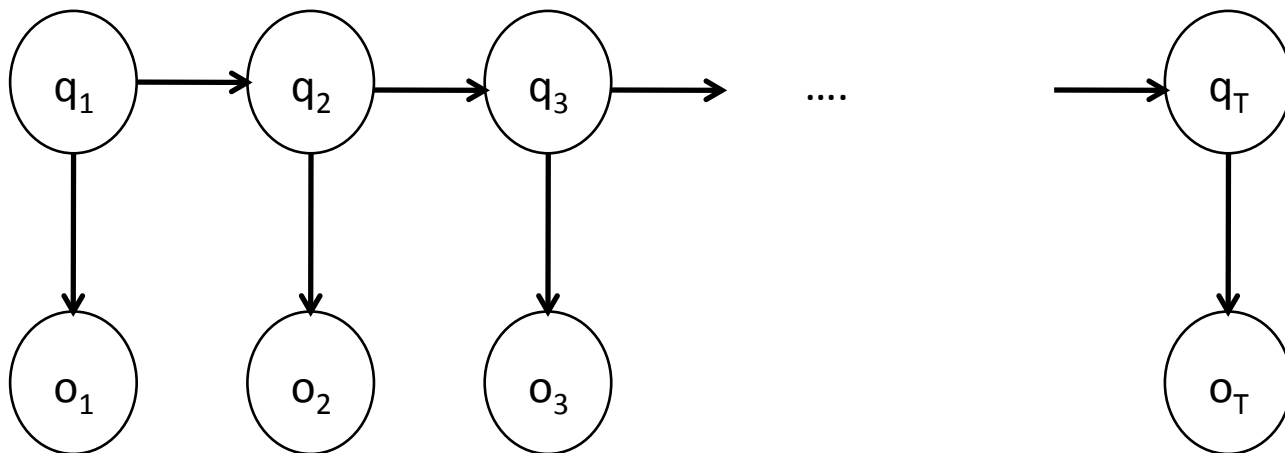
Many of these slides are derived from Tom Mitchell, Ziv Bar-Joseph. Thanks!

# A Hidden Markov Model

- The joint probability of  $(Q,O)$  is defined as

$$P(Q,O) = p(q_1)p(o_1 | q_1) \prod_{t=2}^T p(q_t | q_{t-1})p(o_t | q_t)$$

Initial probability      ↑ transition probability      ↑ emission probability



# Learning HMMs

- Until now we assumed that the emission and transition probabilities are known
- This is usually not the case
  - How is “AI” pronounced by different individuals?
  - What is the probability of hearing “class” after “AI”?

# Learning HMM When Hidden States are Observed

- Assume both hidden and observed states are observed
  - Data:  $((O^1, Q^1), \dots, (O^K, Q^K))$  for  $K$  sequences, where  $O^k = (o_1^k, \dots, o_T^k)$   
 $Q^k = (q_1^k, \dots, q_T^k)$
- MLE for learning!

$$\operatorname{argmax} \log p((O^1, Q^1), \dots, (O^K, Q^K))$$

$$\operatorname{argmax} \log \prod_k p(q_1^k) p(o_1^k | q_1^k) \prod_{t=2}^T p(q_t^k | q_{t-1}^k) p(o_t^k | q_t^k)$$

# Learning HMM When Hidden States are Observed

- MLE for HMM

$$\log p((O^1, Q^1), \dots, (O^K, Q^K))$$

$$= \log \prod_k p(q_1^k) p(o_1^k | q_1^k) \prod_{t=2}^T p(q_t^k | q_{t-1}^k) p(o_t^k | q_t^k)$$

$$= \sum_k \log p(q_1^k) + \sum_k \log p(o_1^k | q_1^k) + \sum_k \sum_t \log p(o_t^k | q_t^k) + \sum_k \sum_t \log p(q_t^k | q_{t-1}^k)$$

Involves only  
initial  
probabilities

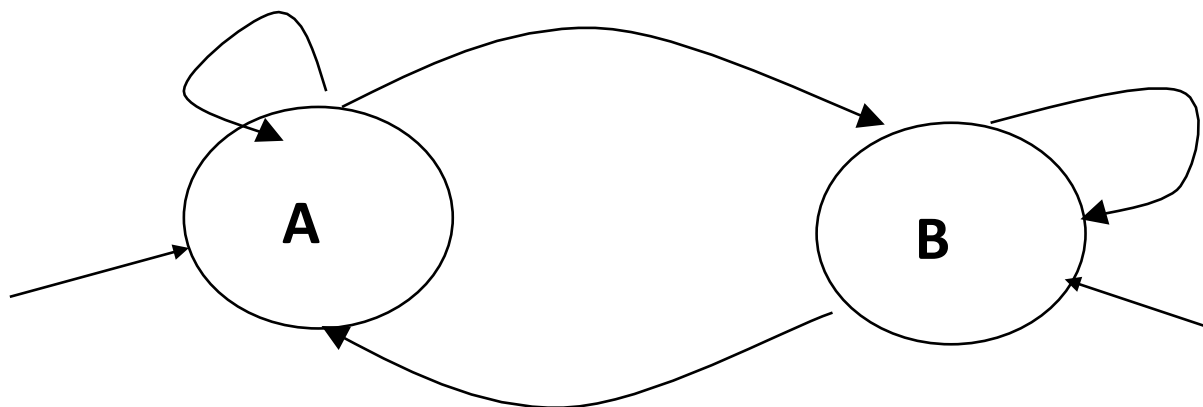
Involves only  
emission  
probabilities

Involves only  
transition  
probabilities

- Differentiate w.r.t. each parameters and set it to 0 and solve!  
Closed form solution

# Example

- Assume the model below
- We also observe the following sequence:  
1,2,2,5,6,5,1  
1,3,2,5,6,5,2  
3,2,1,3,6,5,4
- How can we determine the initial, transition and emission probabilities?



# Initial probabilities

Q: assume we can observe the following sets of states:

AAABBA	1,2,2,5,6,5,1
AABBBBB	1,3,2,5,6,5,2
BAABBAB	3,2,1,3,6,5,4

how can we learn the initial probabilities?

A: Maximum likelihood estimation

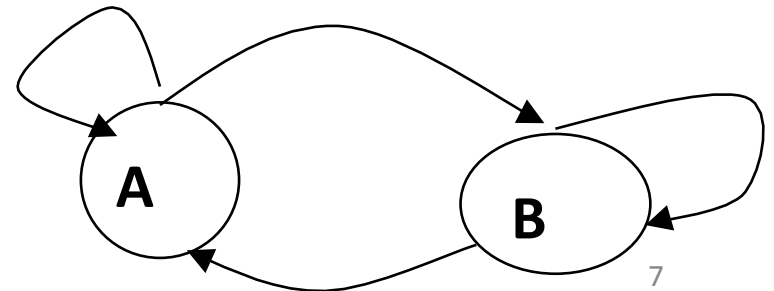
Find the initial probabilities  $\pi$  such that

$$\pi^* = \arg \max \log \prod_k p(q_1^k) p(o_1^k | q_1^k) \prod_{t=2}^T p(q_t^k | q_{t-1}^k) p(o_t^k | q_t^k)$$

$$\pi^* = \arg \max \log \prod_k p(q_1^k)$$

$$\pi_A = \#A / (\#A + \#B)$$

k is the number of sequences available for training



# Transition probabilities

Q: assume we can observe the set of states:

AAABBA	1,2,2,5,6,5,1
AABBBB	1,3,2,5,6,5,2
BAABBAB	3,2,1,3,6,5,4

how can we learn the transition probabilities?

remember that we defined  $a_{i,j} = p(q_t = s_j | q_{t-1} = s_i)$

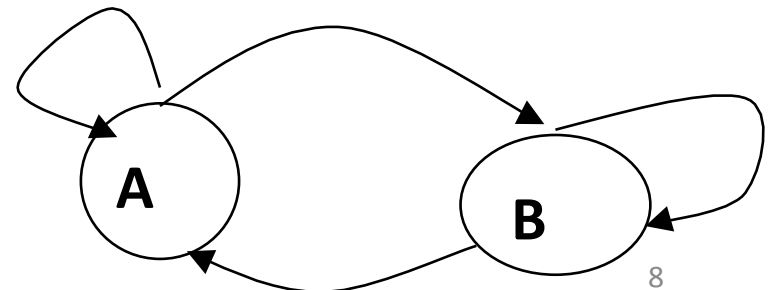
A: Maximum likelihood estimation

Find a transition matrix  $a$  such that

$$a^* = \arg \max_k \log \prod_k p(q_1^k) p(o_1^k | q_1^k) \prod_{t=2}^T p(q_t^k | q_{t-1}^k) p(o_t^k | q_t^k)$$

$$a^* = \arg \max_k \log \prod_k \prod_{t=2}^T p(q_t^k | q_{t-1}^k)$$

$$a_{A,B} = \#AB / (\#AB + \#AA)$$





# Transition probabilities

Q: assume we can observe the set of states:

AAABBA	1,2,2,5,6,5,1
AABBBB	1,3,2,5,6,5,2
BAABBAB	3,2,1,3,6,5,4

Moving window of size 2  
->#AA, #AB, #BA, #BB

how can we learn the transition probabilities?

remember that we defined  
 $a_{i,j} = p(q_t = s_j | q_{t-1} = s_i)$

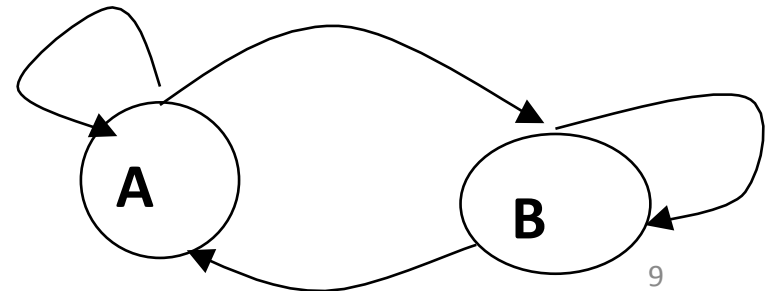
A: Maximum likelihood estimation

Find a transition matrix  $a$  such that

$$a^* = \arg \max_k \log \prod_k p(q_1^k) p(o_1^k | q_1^k) \prod_{t=2}^T p(q_t^k | q_{t-1}^k) p(o_t^k | q_t^k)$$

$$a^* = \arg \max_k \log \prod_k \prod_{t=2}^T p(q_t^k | q_{t-1}^k)$$

$$a_{A,B} = \#AB / (\#AB + \#AA)$$



# Emission probabilities

Q: assume we can observe the set of states:

**AAABBAA** 1,2,2,5,6,5,1

**AABBBBB** 1,3,2,5,6,5,2

**BAABBAB** 3,2,1,3,6,5,4

how can we learn the transition probabilities?

remember that we defined  
 $b_i(o_t) = P(o_t | s_i)$

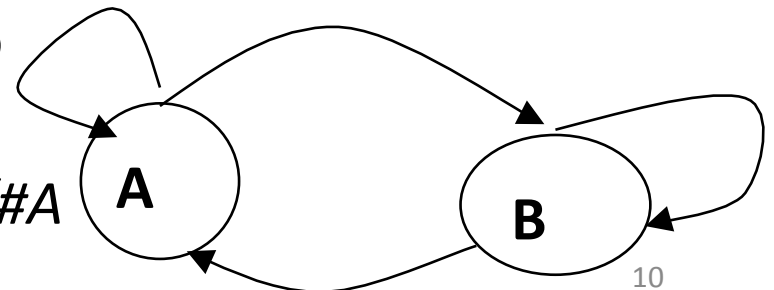
A: Maximum likelihood estimation

Find an emission matrix  $b$  such that

$$b^* = \arg \max \log \prod_k p(q_1^k) p(o_1^k | q_1^k) \prod_{t=2}^T p(q_t^k | q_{t-1}^k) p(o_t^k | q_t^k)$$

$$b^* = \arg \max \log \prod_k p(o_1^k | q_1^k) \prod_{t=2}^T p(o_t^k | q_t^k)$$

$$b_A(5) = \#A5 / (\#A1 + \#A2 + \dots + \#A6) = \#A5 / \#A$$



# Learning HMMs

- In most case we do not know what states generated each of the outputs (hidden states are unobserved)
  - ... but had we known, it would be very easy to determine an emission and transition model!
  - On the other hand, if we had such a model we could determine the set of states using the inference methods we discussed

# Expectation Maximization (EM)

- Appropriate for problems with ‘missing values’ for the variables.
- For example, in HMMs we usually do not observe the states
- Assume complete data log likelihood and maximize **expected log likelihood**

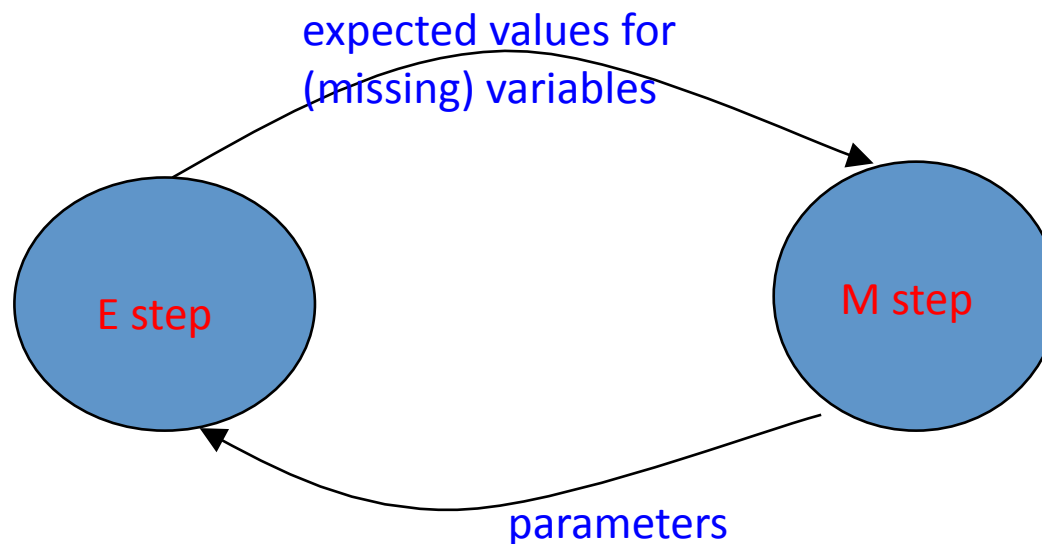
$$\operatorname{argmax} E[\log p((O^1, Q^1), \dots, (O^K, Q^K))]$$

$$\operatorname{argmax} E[\log \prod_k p(q_1^k) p(o_1^k | q_1^k) \prod_{t=2}^T p(q_t^k | q_{t-1}^k) p(o_t^k | q_t^k)]$$

where the expectation is taken with respect to  $p(Q|O, \text{parameters})$

# Expectation Maximization (EM): Quick reminder

- Two steps
  - E step: Fill in the missing variables with the expected values
  - M step: Regular maximum likelihood estimation (MLE) using the values computed in the E step and the values of the other variables
- Guaranteed to converge (though only to a local minima).

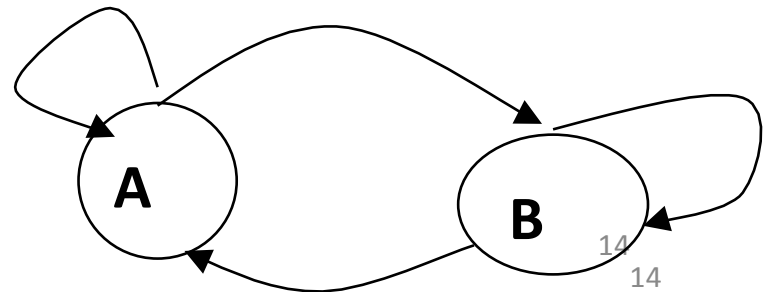


## E Step

- In our example, with complete data, we needed
  - #A, #B to estimate initial probabilities and emission probabilities
  - #AA, #AB, #BA, #BB to estimate transition probabilities
- When hidden states are not observed, we need “expected counts” in E step

$$P(q_t = s_i | O_1, \dots, O_T) = S_t(i)$$

$$P(q_t = s_i, q_{t+1} = s_j | o_1, \dots, o_T) = S_t(i, j)$$



# Forward-Backward

- We already defined a *forward* looking variable

$$\alpha_t(i) = P(O_1 \dots O_t \wedge q_t = s_i)$$

- We also need to define a *backward* looking variable

$$\beta_t(i) = P(O_{t+1}, \dots, O_T \mid s_t = i)$$

# Forward-Backward Algorithm

- We already defined a *forward* looking variable

$$\alpha_t(i) = P(O_1 \dots O_t \wedge q_t = s_i)$$

- We also need to define a *backward* looking variable



# Forward-Backward Algorithm

- Backward step

$$\begin{aligned}\beta_t(i) &= P(O_{t+1}, \dots, O_T \mid q_t = s_i) \\ &= \sum_j P(O_{t+1}, \dots, O_T, q_{t+1} = s_j \mid q_t = s_i) \\ &= \sum_j P(q_{t+1} = s_j \mid q_t = s_i) P(O_{t+1}, \dots, O_T \mid q_{t+1} = s_j, q_t = s_i) \\ &= \sum_j P(q_{t+1} = s_j \mid q_t = s_i) P(O_{t+1}, \dots, O_T \mid q_{t+1} = s_j) \\ &= \sum_j P(q_{t+1} = s_j \mid q_t = s_i) P(O_{t+1} \mid q_{t+1} = s_j) P(O_{t+2}, \dots, O_T \mid q_{t+1} = s_j) \\ &= \sum_j a_{i,j} b_j(O_{t+1}) \beta_{t+1}(j)\end{aligned}$$

# Forward-Backward

- We already defined a *forward* looking variable

$$\alpha_t(i) = P(O_1 \dots O_t \wedge q_t = s_i)$$

- We also need to define a *backward* looking variable

$$\beta_t(i) = P(O_{t+1}, \dots, O_T \mid q_t = s_i)$$

- Using these two definitions we can show

$$P(q_t = s_i \mid O_1, \dots, O_T) \stackrel{\text{def}}{=} \frac{\alpha_t(i)\beta_t(i)}{\sum_j \alpha_t(j)\beta_t(j)} = S_t(i)$$

$P(A \mid B) = P(A, B) / P(B)$

# Forward-Backward

- *forward* looking variable  $\alpha_t(i) = P(O_1 \dots O_t \wedge q_t = s_i)$
- *backward* looking variable  $\beta_t(i) = P(O_{t+1}, \dots, O_T \mid q_t = s_i)$
- Using these two definitions we can show

$$\begin{aligned} P(q_t = s_i \mid O_1, \dots, O_T) &= \frac{P(q_t = s_i, O_1, \dots, O_T)}{P(O_1, \dots, O_T)} \\ &= \frac{P(O_1, \dots, O_t, q_t = s_i) P(O_{t+1}, \dots, O_T \mid q_t = s_i, O_1, \dots, O_t)}{P(O_1, \dots, O_T)} \\ &= \frac{\alpha_t(i) \beta_t(i)}{\sum_j \alpha_t(j) \beta_t(j)} \stackrel{\text{def}}{=} S_t(i) \end{aligned}$$

# State and transition probabilities

- Probability of a state given observations

$$P(q_t = s_i | O_1, \dots, O_T) = \frac{\alpha_t(i)\beta_t(i)}{\sum_j \alpha_t(j)\beta_t(j)} \stackrel{def}{=} S_t(i)$$

- We can also derive a transition probability given observations

$$\begin{aligned} & P(q_t = s_i, q_{t+1} = s_j | o_1, \dots, o_T) \\ &= \frac{\alpha_t(i)P(q_{t+1} = s_j | q_t = s_i)P(o_{t+1} | q_{t+1} = s_j)\beta_{t+1}(j)}{\sum_j \alpha_t(j)\beta_t(j)} \stackrel{def}{=} S_t(i, j) \end{aligned}$$

## E step

- Compute  $S_t(i)$  and  $S_t(i,j)$  for all  $t, i$ , and  $j$  ( $1 \leq t \leq n, 1 \leq i \leq k, 2 \leq j \leq k$ )

$$P(q_t = s_i \mid O_1, \dots, O_T) = S_t(i)$$

$$P(q_t = s_i, q_{t+1} = s_j \mid o_1, \dots, o_T) = S_t(i, j)$$

## M step (1)

Compute transition probabilities:

$$a_{i,j} = \frac{\hat{n}(i, j)}{\sum_k \hat{n}(i, k)}$$

where

$$\hat{n}(i, j) = \sum_t S_t(i, j)$$

## M step (2)

Compute emission probabilities (here we assume a multinomial distribution):

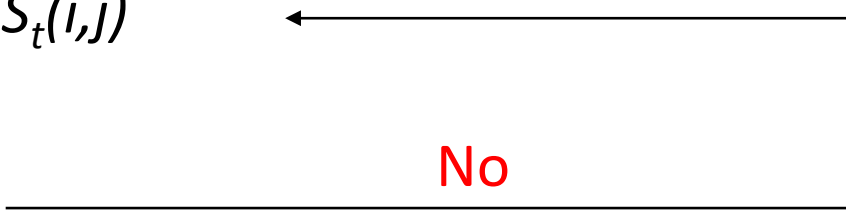
define:

$$B_k(j) = \sum_{t|o_t=j} S_t(k)$$

then

$$b_k(j) = \frac{B_k(j)}{\sum_i B_k(i)}$$

# Complete EM algorithm for learning the parameters of HMMs (Baum-Welch)

- Inputs: 1. Observations  $O_1 \dots O_T$   
2. Number of states, model
1. Guess initial transition and emission parameters
  2. Compute E step:  $S_t(i)$  and  $S_t(i,j)$
  3. Compute M step
  4. Convergence? 
  5. Output complete model

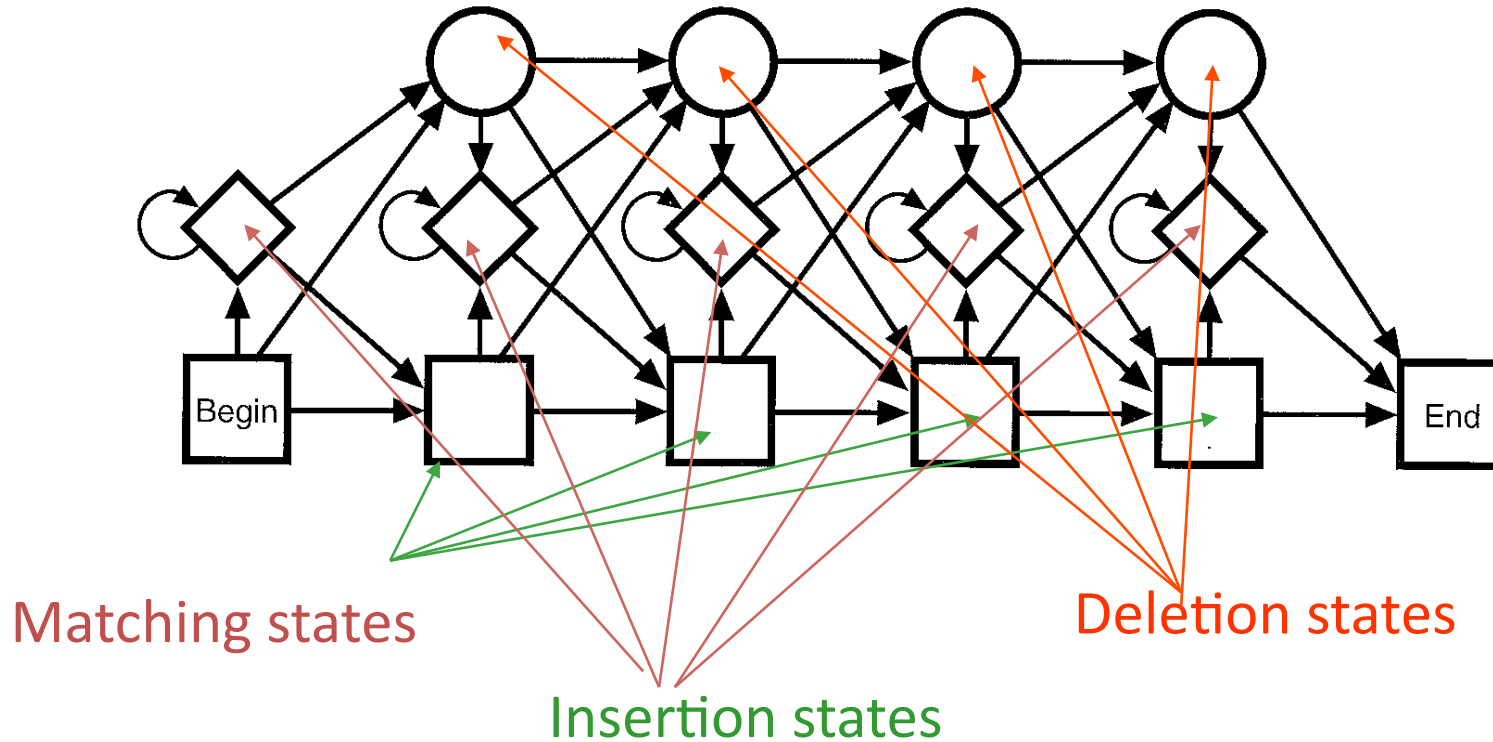
We did not discuss initial probability estimation. These can be deduced from the 1<sup>st</sup> observation in each of the multiple sequences of observations



# States in HMM

- How to decide on the number of states in HMM
  - More states means a more complex model, overfitting!
  - Cross validation
  - Nonparametric Bayesian model

# Building HMMs—Topology

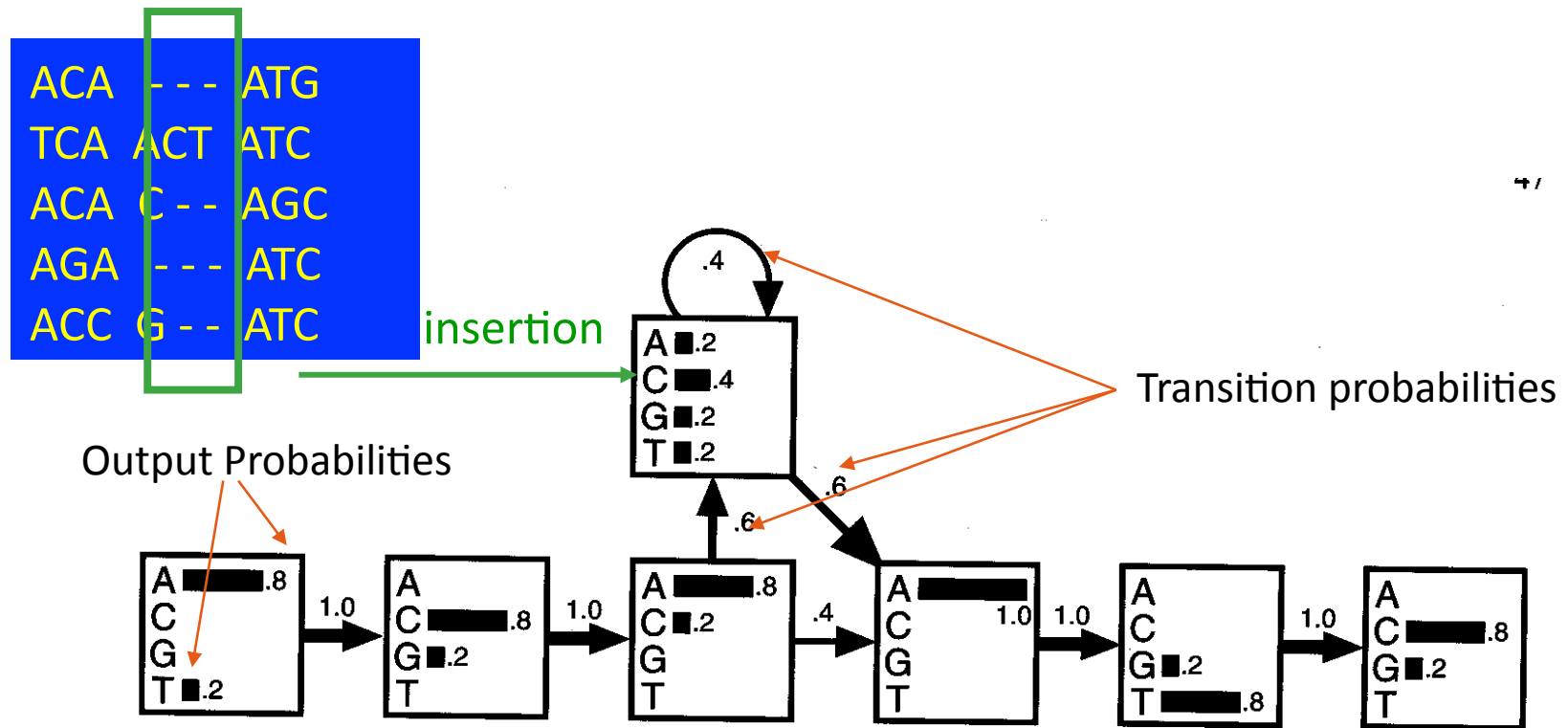


No of matching states = average sequence length in the family

PFAM Database - of Protein families  
<http://pfam.wfam.edu>

M	Q	P	I	L	L	L	V
M	L	R	-	L	L	-	-
M	K	-	I	L	L	L	-
M	P	P	V	L	I	L	V

# Building – *from an existing alignment*



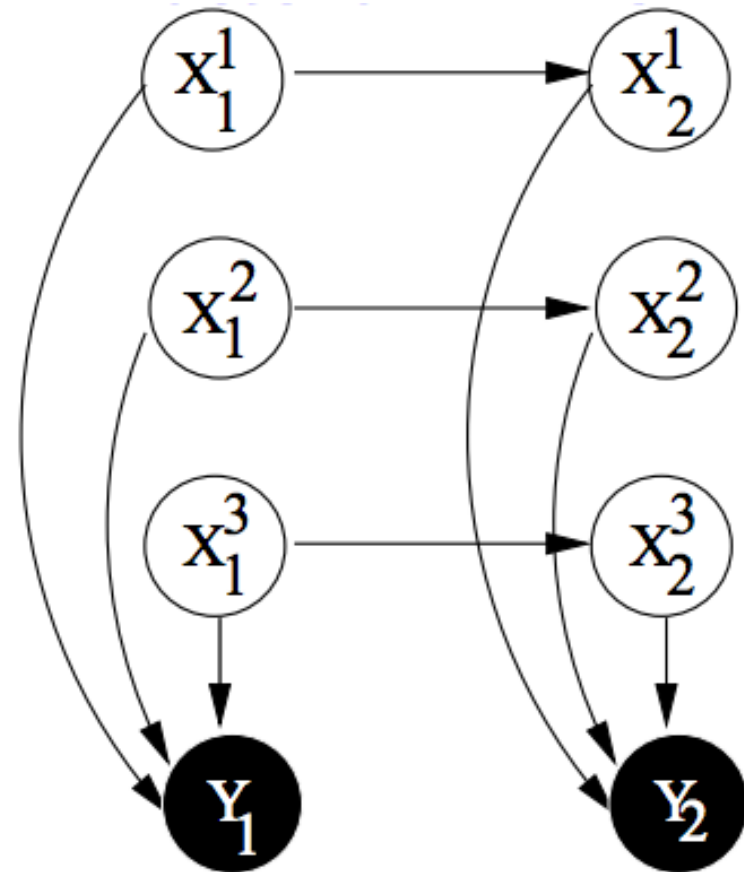
A **HMM model** for a DNA motif alignments, The **transitions** are shown with arrows whose thickness indicate their probability. In each state, the **histogram** shows the probabilities of the four bases.

# Dynamic Bayesian Networks (DBNs)

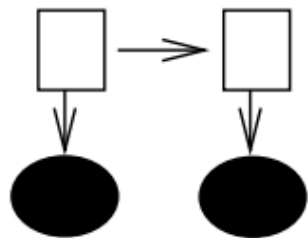
- Bayesian networks for modeling dynamic process. HMM is a special case of DBN
  - HMM represents the state with a single random variable:  $P(Q_t | Q_{t-1})$
  - DBN represents the state with a set of random variables:  $P(Q_t | Q_{t-1})$ , where  $Q_t$  is a set of variables
- DBN often has a compact representation of HMM representations
  - DBN may have exponentially fewer parameters than its corresponding HMM
  - Faster inference and learning

# Factorial HMMs

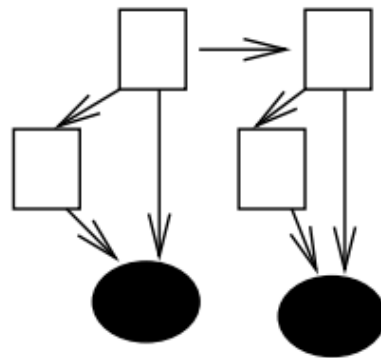
- DBN with  $D$  chains, each with  $K$  states
  - Three  $O(K^2D)$  transition probabilities
  - 12 parameters
- HMM representations?
  - $K^D$  states
  - $O(K^{2D})$  transition probabilities



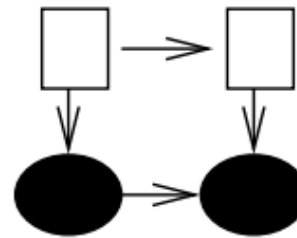
# Other Variants of HMMs as DBNs



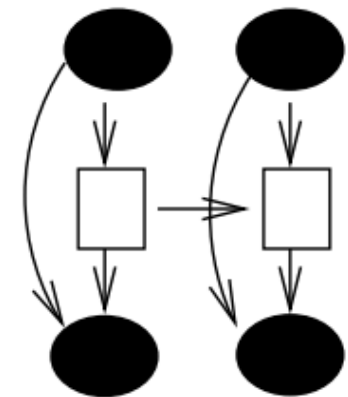
HMM



Mixture of  
Gaussian  
HMM



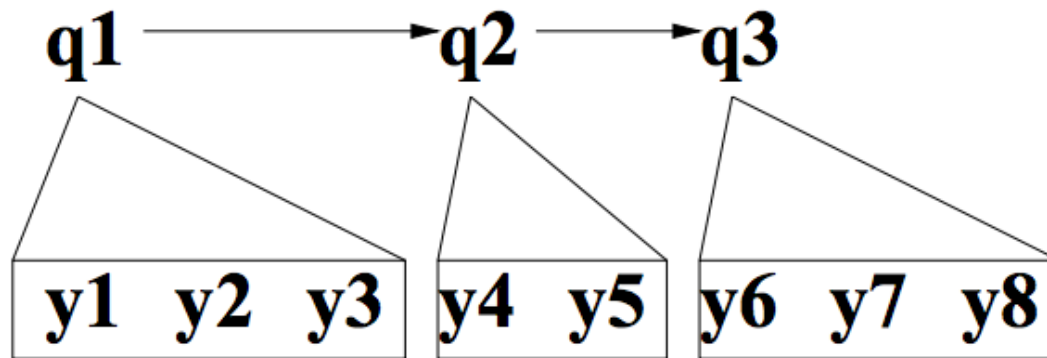
Autoregressive  
HMM



Input-output  
HMM

# Semi-Markov HMM

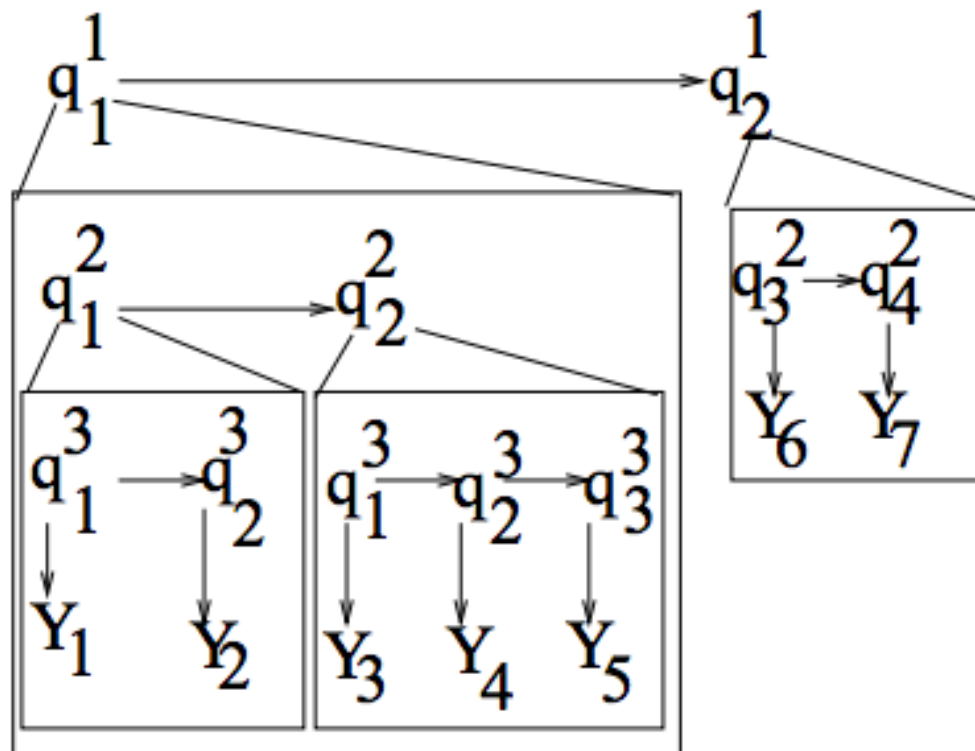
- Relax the Markov constraint to allow staying in the current state for an explicit duration of time  $L_t$



$$P(Y_{t-l+1:l}|Q_t, L_t = l) = \prod_{i=1}^l P(Y_i|Q_t)$$

# Hierarchical HMM

- Each state can emit another HMM that generate sequences





# What you should know

- Why HMMs? Which applications are suitable?
- Inference in HMMs
  - No observations
  - Probability of next state w. observations
  - Maximum scoring path (Viterbi)
- Learning in HMMs
  - EM algorithm with inference as a subroutine