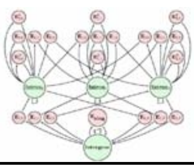
Computational Genomics

10-810/02-710, Spring 2009

Gene Finding and HMM

Eric Xing

Lecture 5, January 28, 2009



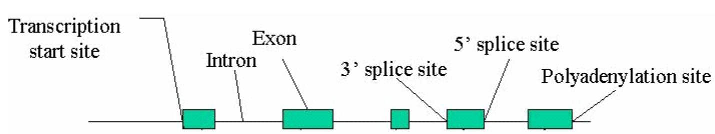
Reading: Durbin Chap 3, class assignment

Probability of a Parse

- What is a parse?

$X_{1:N}$ 124552646214614613613666166466163661636616361651561511514612356234

$y_{1:N}$ - - - - - - - -



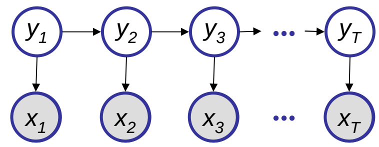
- How to score a parse?

$P_{MAP}(y_t \mid X_{1:N}) = \frac{p(y_t, X_{1:N})}{p(X_{1:N})}$

$P_{HMM}(y_{1:N} \mid X_{1:N}) = p(y_{1:N}, X_{1:N})$

Probability of a Parse

- Given a sequence $\mathbf{x} = x_1 \ldots\ldots x_T$ and a parse $\mathbf{y} = y_1, \ldots\ldots, y_T$,
- To find how likely is the parse: (given our HMM and the sequence)



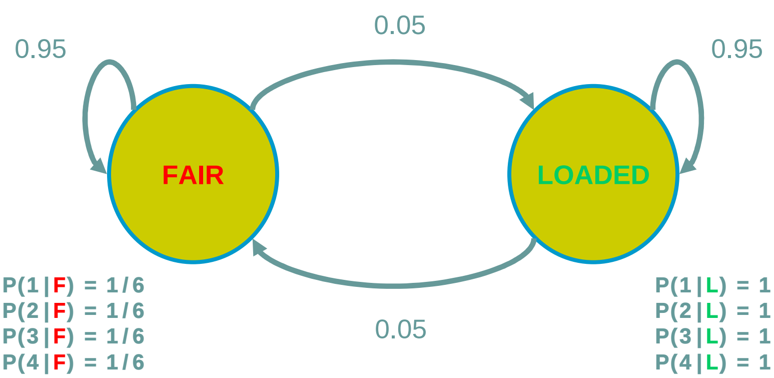
$$\begin{aligned} p(\mathbf{x}, \mathbf{y}) &= p(x_1 \ldots\ldots x_T, y_1, \ldots\ldots, y_T) \qquad \text{(Joint probability)} \\ &= p(y_1)\, p(x_1 \mid y_1)\, p(y_2 \mid y_1)\, p(x_2 \mid y_2) \ldots p(y_T \mid y_{T-1})\, p(x_T \mid y_T) \\ &= p(y_1)\, P(y_2 \mid y_1) \ldots p(y_T \mid y_{T-1}) \times p(x_1 \mid y_1)\, p(x_2 \mid y_2) \ldots p(x_T \mid y_T) \\ &= p(y_1, \ldots\ldots, y_T)\, p(x_1 \ldots\ldots x_T \mid y_1, \ldots\ldots, y_T) \end{aligned}$$

Let $\pi_{y_1} \overset{\text{def}}{=} \prod_{i=1}^{M} [\pi_i]^{y_1^i}$, $\quad a_{y_t, y_{t+1}} \overset{\text{def}}{=} \prod_{i,j=1}^{M} [a_{ij}]^{y_t^i y_{t+1}^j}$, $\quad$ and $\quad b_{y_t, x_t} \overset{\text{def}}{=} \prod_{i=1}^{M} \prod_{k=1}^{K} [b_{ik}]^{y_t^i x_t^k}$,

$$= \pi_{y_1} a_{y_1, y_2} \cdots a_{y_{T-1}, y_T}\; b_{y_1, x_1} \cdots b_{y_T, x_T}$$

- Marginal probability: $p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}) = \sum_{y_1} \sum_{y_2} \cdots \sum_{y_N} \pi_{y_1} \prod_{t=2}^{T} a_{y_{t-1}, y_t} \prod_{t=1}^{T} p(x_t \mid y_t)$
- Posterior probability: $p(\mathbf{y} \mid \mathbf{x}) = p(\mathbf{x}, \mathbf{y}) / p(\mathbf{x})$

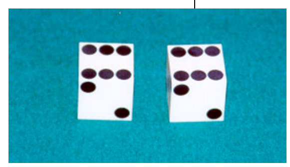
The Dishonest Casino Model



P(1|F) = 1/6
P(2|F) = 1/6
P(3|F) = 1/6
P(4|F) = 1/6
P(5|F) = 1/6
P(6|F) = 1/6

P(1|L) = 1/10
P(2|L) = 1/10
P(3|L) = 1/10
P(4|L) = 1/10
P(5|L) = 1/10
P(6|L) = 1/2

Example: the Dishonest Casino



- Let the sequence of rolls be:
  - x = 1, 2, 1, 5, 6, 2, 1, 6, 2, 4
- Then, what is the likelihood of
  - y = Fair, Fair, Fair, Fair, Fair, Fair, Fair, Fair, Fair, Fair?

    (say initial probs $a_{0Fair} = ½$, $a_{0Loaded} = ½$)

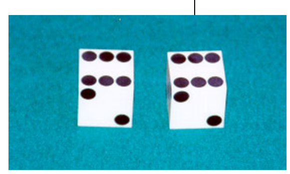
$P(x,y) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2)\cdot$

½ × P(1 | Fair) P(Fair | Fair) P(2 | Fair) P(Fair | Fair) … P(4 | Fair) =

1/6 0.95

$½ \times (\mathbf{1/6})^{10} \times (\mathbf{0.95})^9 = .00000000521158647211 = 5.21 \times 10^{-9}$

Example: the Dishonest Casino



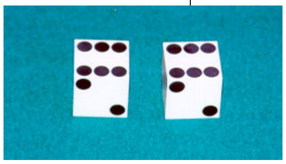
- So, the likelihood the die is fair in all this run is just 5.21×10^{-9}
- OK, but what is the likelihood of
  - π = Loaded, Loaded, Loaded, Loaded, Loaded, Loaded, Loaded, Loaded, Loaded, Loaded?

½ × P(1 | Loaded) P(Loaded | Loaded) … P(4 | Loaded) =

$½ \times (\mathbf{1/10})^8 \times (\mathbf{1/2})^2 (\mathbf{0.95})^9 = .00000000078781176215 = 0.79 \times 10^{-9}$

- Therefore, it is after all 6.59 times more likely that the die is fair all the way, than that it is loaded all the way

Example: the Dishonest Casino



- Let the sequence of rolls be:
  - x = 1, 6, 6, 5, 6, 2, 6, 6, 3, 6
- Now, what is the likelihood π = F, F, …, F?
  - $\frac{1}{2} \times (1/6)^{10} \times (0.95)^9 = 0.5 \times 10^{-9}$, same as before
- What is the likelihood y = L, L, …, L?

$\frac{1}{2} \times (1/10)^4 \times (1/2)^6 (0.95)^9 = .00000049238235134735 = 5 \times 10^{-7}$

- So, it is 100 times more likely the die is loaded

Applications of HMMs

- **Some early applications of HMMs**
  - finance, but we never saw them
  - speech recognition
  - modelling ion channels
- In the mid-late 1980s HMMs entered genetics and molecular biology, and they are now firmly entrenched.
- **Some current applications of HMMs to biology**
  - mapping chromosomes
  - aligning biological sequences
  - predicting sequence structure
  - inferring evolutionary relationships
  - finding genes in DNA sequence

The HMM Algorithms

Questions:

- **Decoding**: What is the most likely DNA parsing? Viterbi
- **Evaluation**: What is the probability of the observed sequence? Forward
- **Decoding**: What is the probability that the state of the 3rd position is Bk or gene, given the observed sequence? Forward-Backward
- **Learning**: Under what parameterization are the observed sequences most probable? Baum-Welch (EM)

Decoding

- GIVEN $\mathbf{x} = x_1, \ldots, x_T$, we want to find $\mathbf{y} = y_1, \ldots, y_T$, such that $P(\mathbf{y}|\mathbf{x})$ is maximized:

$$\mathbf{y}^* = \text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) = \text{argmax}_{\pi} P(\mathbf{y},\mathbf{x})$$

$$p(y|x) = \frac{p(x,y)}{p(x)}$$

Viterbi decoding

- GIVEN $\mathbf{x} = x_1, \ldots, x_T$, we want to find $\mathbf{y} = y_1, \ldots, y_T$, such that $P(\mathbf{y}|\mathbf{x})$ is maximized:

$$\mathbf{y}^* = \text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) = \text{argmax}_{\pi} P(\mathbf{y},\mathbf{x})$$

$V_T^k : ?\ \forall k.$

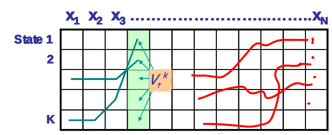
  - Let

$$V_t^k = \max_{\{y_1,\ldots,y_{t-1}\}} P(x_1,\ldots,x_{t-1}, y_1,\ldots,y_{t-1}, x_t, y_t^k = 1)$$

= Probability of most likely **sequence of states** ending at state $y_t = k$

  - The recursion:

$$V_t^k = p(x_t \mid y_t^k = 1) \max_i a_{i,k} V_{t-1}^i$$



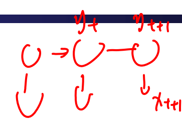
  - Underflows are a significant problem

$$p(x_1,\ldots,x_t,y_1,\ldots,y_t) = \pi_{y_1} a_{y_1,y_2} \cdots a_{y_{t-1},y_t} b_{y_1,x_1} \cdots b_{y_t,x_t}$$

    - These numbers become extremely small – underflow
    - Solution: Take the logs of all values: $V_t^k = \log p(x_t \mid y_t^k = 1) + \max_i \left(\log(a_{i,k}) + V_{t-1}^i\right)$

The Viterbi Algorithm – derivation

- Define the viterbi probability:



$$V_{t+1}^k = \max_{\{y_1,\ldots,y_t\}} P(x_1,\ldots,x_t, y_1,\ldots,y_t, x_{t+1}, y_{t+1}^k = 1)$$

$$= \max_{\{y_1,\ldots,y_t\}} P(x_{t+1}, y_{t+1}^k = 1 \mid x_1,\ldots,x_t, y_1,\ldots,y_t) P(x_1,\ldots,x_t, y_1,\ldots,y_t)$$

$$= \max_{\{y_1,\ldots,y_t\}} P(x_{t+1}, y_{t+1}^k = 1 \mid y_t) P(x_1,\ldots,x_{t-1}, y_1,\ldots,y_{t-1}, x_t, y_t)$$

$$= \max_i P(x_{t+1}, y_{t+1}^k = 1 \mid y_t^i = 1) \max_{\{y_1,\ldots,y_{t-1}\}} P(x_1,\ldots,x_{t-1}, y_1,\ldots,y_{t-1}, x_t, y_t^i = 1)$$

$$= \max_i P(x_{t+1} \mid y_{t+1}^k = 1) a_{i,k} V_t^i$$

V_t^i

$$= P(x_{t+1} \mid y_{t+1}^k = 1) \max_i a_{i,k} V_t^i$$

$P(x_{t+1}, y_{t+1} \mid y_t)$

$= P(x_{t+1} \mid y_{t+1}) P(y_{t+1} \mid y_t)$

$=$

The Viterbi Algorithm

- Input: $\mathbf{x} = x_1, \ldots, x_T,$

$$V_t^k = p(x_t \mid y_t^k = 1)\max_i a_{i,k} V_{t-1}^i$$

Initialization:

$$V_1^k = P(x_1 \mid y_1^k = 1)\pi_k$$

Iteration:

$$V_t^k = P(x_t \mid y_t^k = 1)\max_i a_{i,k} V_{t-1}^i$$

$$\mathrm{Ptr}(k,t) = \arg\max_i a_{i,k} V_{t-1}^i$$

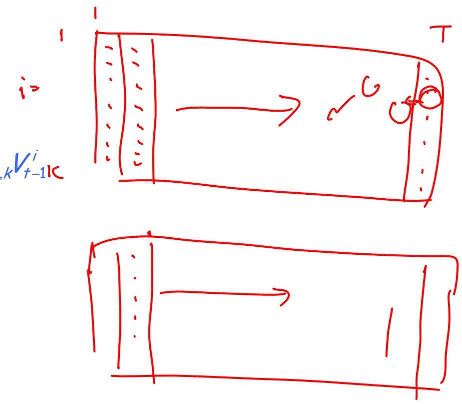
Termination:

$$P(\mathbf{x}, \mathbf{y}^*) = \max_k V_T^k$$

TraceBack:

$$y_T^* = \arg\max_k V_T^k$$

$$y_{t-1}^* = \mathrm{Ptr}(y_t^*, t)$$



$O(k^2T)$

The HMM Algorithms

Questions:

- **Decoding**: What is the most likely DNA parsing? Viterbi
- **Evaluation**: What is the probability of the observed sequence? Forward
- **Decoding**: What is the probability that the state of the 3rd position is Bk or gene, given the observed sequence? Forward-Backward
- **Learning**: Under what parameterization are the observed sequences most probable? Baum-Welch (EM)

The likelihood of a sequence

- We want to calculate $P(\mathbf{x})$, the likelihood of $\mathbf{x}$, given the HMM
  - Sum over all possible ways of generating $\mathbf{x}$:

$$p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}) = \sum_{y_1} \sum_{y_2} \cdots \sum_{y_N} \pi_{y_1} \prod_{t=2}^{T} a_{y_{t-1}, y_t} \prod_{t=1}^{T} p(x_t \mid y_t)$$

- Complexity?

$$O(K^T)$$

- Why useful?

$$p(y_t \mid x) = \frac{p(y_t, x)}{p(x)}$$

The Forward Algorithm

- We want to calculate $P(\mathbf{x})$, the likelihood of $\mathbf{x}$, given the HMM
  - Sum over all possible ways of generating $\mathbf{x}$:

$$p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}) = \sum_{y_1} \sum_{y_2} \cdots \sum_{y_N} \pi_{y_1} \prod_{t=2}^{T} a_{y_{t-1}, y_t} \prod_{t=1}^{T} p(x_t \mid y_t)$$

  - To avoid summing over an exponential number of paths $\mathbf{y}$, define

$$\alpha(y_t^k = 1) = \alpha_t^k \overset{\text{def}}{=} P(x_1, \ldots, x_t, y_t^k = 1)$$

(the forward probability)

$\alpha_T^k \quad \forall k.$

$= p(x_1 \cdots x_T, y_T^k = 1)$

$p(x) = \sum_k p(x, y_T^k = 1)$

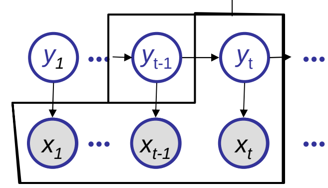
  - The recursion:

$$\alpha_t^k = p(x_t \mid y_t^k = 1) \sum_i \alpha_{t-1}^i a_{i,k}$$

$$P(\mathbf{x}) = \sum_k \alpha_T^k$$

The Forward Algorithm – derivation

- Compute the forward probability:



$$\begin{aligned}
\alpha_t^k &= P(x_1,\ldots,x_{t-1},x_t,y_t^k=1) \\
&= \sum_{y_{t-1}} P(x_1,\ldots,x_{t-1},x_t,y_{t-1},y_t^k=1) \\
&= \sum_{y_{t-1}} P(x_1,\ldots,x_{t-1},y_{t-1})P(y_t^k=1\mid y_{t-1},x_1,\ldots,x_{t-1})P(x_t\mid y_t^k=1,x_1,\ldots,x_{t-1},y_{t-1}) \\
&= \sum_{y_{t-1}} P(x_1,\ldots,x_{t-1},y_{t-1})P(y_t^k=1\mid y_{t-1})P(x_t\mid y_t^k=1) \\
&= P(x_t\mid y_t^k=1)\sum_i P(x_1,\ldots,x_{t-1},y_{t-1}^i=1)P(y_t^k=1\mid y_{t-1}^i=1) \\
&= P(x_t\mid y_t^k=1)\sum_i \alpha_{t-1}^i a_{i,k}
\end{aligned}$$

α_{t-1}^i?

Chain rule: $P(A,B,C)=P(A)P(B\mid C)P(C\mid A,B)$

The Forward Algorithm

$V_t^k = P(x_t|y) \max_i a_{i,k} V_{t-1}$

- We can compute α_t^k for all k, t, using dynamic programming!

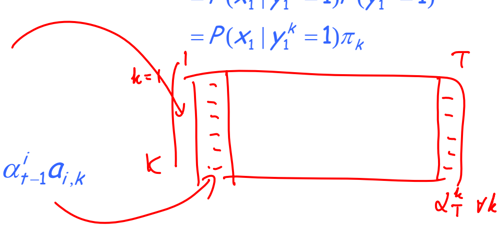
Initialization:

$$\alpha_1^k = P(x_1\mid y_1^k=1)\pi_k$$

$$\begin{aligned}
\alpha_1^k &= P(x_1,y_1^k=1) \\
&= P(x_1\mid y_1^k=1)P(y_1^k=1) \\
&= P(x_1\mid y_1^k=1)\pi_k
\end{aligned}$$

Iteration:

$$\alpha_t^k = P(x_t\mid y_t^k=1)\sum_i \alpha_{t-1}^i a_{i,k}$$



Termination:

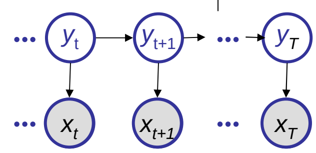
$$P(\mathbf{x}) = \sum_k \alpha_T^k$$

The Backward Algorithm

$P(x)$

- We want to compute $P(y_t^k = 1 \mid \mathbf{x})$,

  the posterior probability distribution on the t^{th} position, given $\mathbf{x}$



  - We start by computing

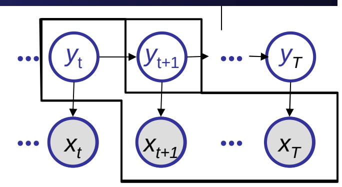
$$P(y_t^k = 1, \mathbf{x}) = P(x_1, \ldots, x_t, y_t^k = 1, x_{t+1}, \ldots, x_T)$$
$$= P(x_1, \ldots, x_t, y_t^k = 1) P(x_{t+1}, \ldots, x_T \mid x_1, \ldots, x_t, y_t^k = 1)$$
$$= \underbrace{P(x_1 \ldots x_t, y_t^k = 1)}_{\text{Forward, } \alpha_t^k} \underbrace{P(x_{t+1} \ldots x_T \mid y_t^k = 1)}_{\text{Backward, } \beta_t^k = P(x_{t+1}, \ldots, x_T \mid y_t^k = 1)}$$

  - The recursion:

$$\beta_t^k = \sum_i a_{k,i} p(x_{t+1} \mid y_{t+1}^i = 1) \beta_{t+1}^i$$

The Backward Algorithm – derivation

- Define the backward probability:



$$\beta_t^k = P(x_{t+1}, \ldots, x_T \mid y_t^k = 1)$$
$$= \sum_{y_{t+1}} P(x_{t+1}, \ldots, x_T, y_{t+1} \mid y_t^k = 1)$$
$$= \sum_i P(y_{t+1}^i = 1 \mid y_t^k = 1) p(x_{t+1} \mid y_{t+1}^i = 1, y_t^k = 1) P(x_{t+2}, \ldots, x_T \mid x_{t+1}, y_{t+1}^i = 1, y_t^k = 1)$$
$$= \sum_i P(y_{t+1}^i = 1 \mid y_t^k = 1) p(x_{t+1} \mid y_{t+1}^i = 1) P(x_{t+2}, \ldots, x_T \mid y_{t+1}^i = 1)$$
$$= \sum_i a_{k,i} p(x_{t+1} \mid y_{t+1}^i = 1) \beta_{t+1}^i$$

Chain rule: $P(A, B, C \mid \alpha) = P(A, \alpha) P(B \mid C, \alpha) P(C \mid A, B, \alpha)$

The Backward Algorithm

$P(y_t^k \mid X)$

$= \frac{P(y_t^k, X)}{P(X)}$

$= \frac{\alpha_t^k \beta_t^k}{\sum \alpha_t^k \beta_t^k}$

- We can compute β_t^k for all k, t, using dynamic programming!

Initialization:

$$\beta_T^k = 1, \ \forall k$$

Iteration:

$$\beta_t^k = \sum_i a_{k,i} P(x_{t+1} \mid y_{t+1}^i = 1) \beta_{t+1}^i$$

Termination:

$$P(\mathbf{x}) = \sum_k \alpha_1^k \beta_1^k$$

$P(X, y_t^k) = \alpha_t^k \beta_t^k$

$P(X) = \sum_{y_t^k} \ldots$

Example:

x = 1, 2, 1, 5, 6, 2, 1, 6, 2, 4

$\alpha_1^1 = \pi_1 P(X=1 \mid y_1^1 = 1) = 0.5 \times 1/6 = 0.0833$

$\alpha_1^2 = \pi_2 P(X=1 \mid y_1^2 = 1) = 0.5 \times 1/10$

$\alpha_2^1 = \frac{1}{6} \times [\alpha_1^1 \times a_{1,1} + \alpha_1^2 \times a_{2,1}]$

$= \frac{1}{6} \times 0.0833 \times 0.95 + 0.05 \times 0.05]$

$= \ldots$

P(1|F) = 1/6 P(1|L) = 1/10
P(2|F) = 1/6 P(2|L) = 1/10
P(3|F) = 1/6 P(3|L) = 1/10
P(4|F) = 1/6 P(4|L) = 1/10
P(5|F) = 1/6 P(5|L) = 1/10
P(6|F) = 1/6 P(6|L) = 1/2

$$\alpha_t^k = P(x_t \mid y_t^k = 1) \sum_i \alpha_{t-1}^i a_{i,k}$$

$$\beta_t^k = \sum_i a_{k,i} P(x_{t+1} \mid y_{t+1}^i = 1) \beta_{t+1}^i$$

$\beta_T^1 = 1$

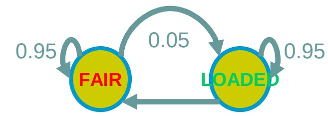
$\beta_T^2 = 1$

$\beta_{T-1}^1 =$

$\beta_{T-1}^2 =$

x = 1, 2, 1, 5, 6, 2, 1, 6, 2, 4

Alpha (actual)		Beta (actual)	
0.0833	0.0500	0.0000	0.0000
0.0136	0.0052	0.0000	0.0000
0.0022	0.0006	0.0000	0.0000
0.0004	0.0001	0.0000	0.0000
0.0001	0.0000	0.0001	0.0001
0.0000	0.0000	0.0007	0.0006
0.0000	0.0000	0.0045	0.0055
0.0000	0.0000	0.0264	0.0112
0.0000	0.0000	0.1633	0.1033
0.0000	0.0000	1.0000	1.0000



P(1|F) = 1/6
P(2|F) = 1/6
P(3|F) = 1/6
P(4|F) = 1/6
P(5|F) = 1/6
P(6|F) = 1/6

P(1|L) = 1/10
P(2|L) = 1/10
P(3|L) = 1/10
P(4|L) = 1/10
P(5|L) = 1/10
P(6|L) = 1/2

$$\alpha_t^k = P(x_t \mid y_t^k = 1) \sum_i \alpha_{t-1}^i a_{i,k}$$

$$\beta_t^k = \sum_i a_{k,i} P(x_{t+1} \mid y_{t+1}^i = 1) \beta_{t+1}^i$$

x = 1, 2, 1, 5, 6, 2, 1, 6, 2, 4   log α

Alpha (logs)		Beta (logs)	
-2.4849	-2.9957	-16.2439	-17.2014
-4.2969	-5.2655	-14.4185	-14.9922
-6.1201	-7.4896	-12.6028	-12.7337
-7.9499	-9.6553	-10.8042	-10.4389
-9.7834	-10.1454	-9.0373	-9.7289
-11.5905	-12.4264	-7.2181	-7.4833
-13.4110	-14.6657	-5.4135	-5.1977
-15.2391	-15.2407	-3.6352	-4.4938
-17.0310	-17.5432	-1.8120	-2.2698
-18.8430	-19.8129	0	0

$P(y_5^1 = 1 \mid x)$
$+ P(y_5^0 = 1 \mid x) = 1$

P(1|F) = 1/6
P(2|F) = 1/6
P(3|F) = 1/6
P(4|F) = 1/6
P(5|F) = 1/6
P(6|F) = 1/6

P(1|L) = 1/10
P(2|L) = 1/10
P(3|L) = 1/10
P(4|L) = 1/10
P(5|L) = 1/10
P(6|L) = 1/2

$$\alpha_t^k = P(x_t \mid y_t^k = 1) \sum_i \alpha_{t-1}^i a_{i,k}$$

$$\beta_t^k = \sum_i a_{k,i} P(x_{t+1} \mid y_{t+1}^i = 1) \beta_{t+1}^i$$

$$P(y_5^1 \mid x) = \frac{\alpha_5^1 \beta_5^1}{P(x)} = \frac{\exp(\log \alpha_5^1 + \log \beta_5^1)}{P(x)}$$

$$= \frac{\exp(-9.7834 - 9.0373)}{P(x)}$$

What is the probability of a hidden state prediction?

$\boldsymbol{x}$ = 1, 2, 1, 5, 6, 2, 1, 6, 2, 4

$$P(y_5^1|x) = \frac{\exp(-18.82)}{P(x)}$$

$$P(y_5^2|x) = \frac{\exp(-19.8743)}{P(x)}$$

$= 1$

$$P(y_5^1|x) = \frac{P(y_5^1=1|x)}{P(y_5^1=1|x)+P(y_5^2=1|x)}$$

$$= \frac{\exp(-18.82)}{\exp(-18.82)+\exp(-19.8743)} = 0.7445$$

$$P(y_8^1|x) = 0.657$$

Posterior decoding

- We can now calculate

$$P(y_t^k = 1 | \mathbf{x}) = \frac{P(y_t^k = 1, \mathbf{x})}{P(\mathbf{x})} = \frac{\alpha_t^k \beta_t^k}{P(\mathbf{x})}$$

- Then, we can ask
  - What is the most likely state at position t of sequence $\mathbf{x}$:

$$k_t^* = \arg\max_k P(y_t^k = 1 | \mathbf{x})$$

  - Note that this is an MPA of a single hidden state, what if we want to a MPA of a whole hidden state sequence?
  - Posterior Decoding: $\left\{ y_t^{k_t^*} = 1 : t = 1 \cdots T \right\}$
  - This is different from MPA of a whole sequence of hidden states
  - This can be understood as *bit error rate* vs. *word error rate*

Example:
MPA of *X* ?
MPA of *(X, Y)* ?

x	y	$P(x,y)$
0	0	0.35
0	1	0.05
1	0	0.3
1	1	0.3

Computational Complexity and implementation details

- What is the running time, and space required, for Forward, and Backward?

$$\alpha_t^k = p(x_t \mid y_t^k = 1)\sum_i \alpha_{t-1}^i a_{i,k}$$
$$\beta_t^k = \sum_i a_{k,i} p(x_{t+1} \mid y_{t+1}^i = 1)\beta_{t+1}^i$$
$$V_t^k = p(x_t \mid y_t^k = 1)\max_i a_{i,k} V_{t-1}^i$$

Time: $O(K^2 N)$; Space: $O(KN)$.

- Useful implementation technique to avoid underflows
  - Viterbi: sum of logs
  - Forward/Backward: rescaling at each position by multiplying by a constant