Finite state machine

An alignment corresponds to a path in the finite state machine.
Probabilistic model HMM for related sequences
With transitional probabilities and emission probabilities, generates two sequences simultaneously

Modeling length
By adding an end state

Example
x = VLSPADK    y = HLAESK
Possible alignment:

What is the most probable alignment (path in HMM)?
Viterbi algorithm

- All the algorithms we have seen for HMM apply, e.g. Viterbi
- But there is an extra dimension in the search space because of the extra emitted sequence
- Instead of using $v_{ij}(j)$, we will use $v_{i,j}(j)$ because an observation $x_i$ does not necessarily mean an observation for $y_j$.
- Imagine we have two clocks, one for $x$, and one for $y$, that work differently in different zones.
- Therefore, $v_{i,j}(j)$ can advance only in certain ways:
  - In zone A, both $i$ and $j$ advance.
  - In zone B, only $i$ advances.
  - In zone C, only $j$ advances.

\[
\begin{align*}
v_i(0,0) &= 1, \quad v_i(0,j) = v_i(0,0) = 0 \\
v_i(0,0) &= 0, \quad v_i(1,0) = \alpha_i^0p_{i0}, \quad v_i(0,0) = 0 \\
v_i(0,0) &= 0, \quad v_i(0,0) = 0, \quad v_i(0,0) = \alpha_i^0p_{i0}, \quad v_i(0,0) = 0
\end{align*}
\]

\[
\begin{align*}
&\text{for } i=1\ldots m, j=1\ldots n \\
&v_i(j, j) = P_xj_j, \quad \text{max} \left\{ \begin{array}{l}
(1-c)(v_i(j-1, j-1) \\
(1-a)(v_i(j-1, j-1)
\end{array} \right.
\end{align*}
\]

\[
\begin{align*}
v_i(j, j) &= P_xj_j, \quad \text{max} \left\{ \begin{array}{l}
v_i(j, j) = v_i(j, j) \\
v_i(j, j) = v_i(j, j)
\end{array} \right.
\end{align*}
\]

\[
\begin{align*}
v_i(j, j) &= P_yj_j, \quad \text{max} \left\{ \begin{array}{l}
v_i(j, j) = v_i(j, j) \\
v_i(j, j) = v_i(j, j)
\end{array} \right.
\end{align*}
\]

\[
v_i = \text{max}(v_i(m, n), v_i(m, n), v_i(m, n))
\]

Observation

- Viterbi algorithm looks very similar to Needleman-Wunsch
- Let’s investigate more this similarity
- Performing Viterbi in log space will definitely make it look like Needleman-Wunsch (multiplication $\rightarrow$ additive), let’s do even more…
- Let’s compute the log-odds ratio of the max probability of an alignment (path) to the probability of obtaining the sequences at random, therefore compute the maximum log-odds ratio.
  - Construct a model for random sequences
  - Perform Viterbi in log space with both models in mind
- We should then see almost the same algorithm
Two HMMs at the same time

- Alignment HMM path
  \( \rho = p_1, p_2, p_3 \ldots \)
  for each step here....

- Random HMM path
  \( \rho = q_1, q_2, q_3 \ldots \)
  find a corresponding step here....

- Align steps together

Random model

- \( x \) and \( y \) are generated independently according to the individual probabilities \( P_x \) and \( P_y \)

\[
p(x, y | R) = \mu (1 - \mu) \prod_{t=1}^{n} P_{x_t} \cdot \mu (1 - \mu) \prod_{t=1}^{n} P_{y_t}
\]

Aligned v.s. random (identifying steps)

- In the alignment model:
  - Each match step contributes \( (1 - \mu) P_{x_t} \) (\( \ldots \) depends)
  - Each start gap step contributes \( \mu P_{x_1} \) or \( \mu P_{y_1} \)
  - Each additional gap step contributes \( \delta P_{x_t} \) or \( \delta P_{y_t} \)

- In the random model:
  - Each letter step contributes \( (1 - \mu) P_{x_t} \) or \( (1 - \mu) P_{y_t} \)
  - All terms are counted except for a \( \mu^2 \)
**Aligned v.s. random (associating steps)**

<table>
<thead>
<tr>
<th>Alignment model</th>
<th>Random model</th>
</tr>
</thead>
<tbody>
<tr>
<td>match ( x, y )</td>
<td>generate ( x ) and ( y )</td>
</tr>
<tr>
<td>start gap in ( y ) with ( x )</td>
<td>generate ( x )</td>
</tr>
<tr>
<td>start gap in ( x ) with ( y )</td>
<td>generate ( y )</td>
</tr>
<tr>
<td>gap in ( y ) with ( x )</td>
<td>generate ( x )</td>
</tr>
<tr>
<td>gap in ( x ) with ( y )</td>
<td>generate ( y )</td>
</tr>
</tbody>
</table>

**Obtain \( p/q \) for each step...**

\[
x(x, y) = \log \frac{P_{x,y}}{P_x P_y} = \log \frac{P_{x,y}}{P_x P_y} + \log \frac{1 - 2x - \tau}{(1 - \mu)^2} + \log \frac{1 - 2x - \tau}{(1 - \mu)^2} = \log \frac{P_{x,y}}{P_x P_y}.
\]

- match step
- corresponding random step
- gap step
- corresponding random step
- gap step
- corresponding random step
- gap step
- corresponding random step

\[
x(x, y) = \log \frac{1 - \delta - \tau}{(1 - \mu)^2} = \log \frac{1 - \delta - \tau}{(1 - \mu)^2} = -\log \delta
\]

- \( x = -\log \left( \frac{1 - \delta - \tau}{(1 - \mu)^2} \right) \)

- adjustment factor to correct for \( \log \delta \)

**Start gap adjustment factor**

- The contribution of state A assumes that a match always follows a match.
- If state A (match) is reached from state B or state C (a gap), then the term contributed to the probability of the path is \((1-\mu)/P_x P_y\), and not \((1-2\mu)/P_x P_y\).
- We correct this before it happens when we first open the gap (go from A to B or C) by multiplying \((1-\mu)(1-2\mu)\) by the factor to correct for \( \log \delta \).
- But what if we never go back to A? Then the score does not exactly correspond to the log-odds ratio.
**log-odds ratio Viterbi (Needleman-Wunsch)**

\[ \nu_d(0,0) = 2 \log q, \quad \nu_d(0,1) = \nu_d(0), \quad \nu_d(0,1) = - \nu_d(0) \]

\[ \nu_d(0,0) = - \nu_d(0), \quad \nu_d(0,1) = - \nu_d(0) \]

for \( i = 1 \ldots m, \quad j = 1 \ldots n \)

\[ \nu_d(i,j) = s(x_i,y_j) + \max \begin{cases} \nu_d(i-1,j-1) \nu_d(i-1,j-1) \\ \nu_d(i-1,j-1) \nu_d(i-1,j-1) \end{cases} \]

\[ \nu_d(i,j) = \max \begin{cases} \nu_d(i-1,j) \nu_d(i-1,j) \\ \nu_d(i,j-1) \nu_d(i,j-1) \end{cases} \]

\[ \nu_d(i,j) = \max \begin{cases} \nu_d(i-1,j) \nu_d(i-1,j) \\ \nu_d(i,j-1) \nu_d(i,j-1) \end{cases} \]

\[ \nu_d = \max(\nu_d(m,n), \nu_d(m,n) - c, \nu_d(m,n) - c) \]

\[ c = \log(1-\gamma) - \log(1-2\gamma) \]

---

**What is the meaning of the scores?**

- From Viterbi, we obtained Needleman-Wunsch with scores directly related to the parameters of a probabilistic model.

  Needelmann-Wunsch is just Viterbi with an appropriate transformation!

- We knew from before that the score is related to the log-odds ratio of probabilities. Now Viterbi gives more justification for this (we are actually after the most likely alignment compared to just a random instance).

- We have a clear relation between the scores (even for gaps now) of Needleman-Wunsch and the parameters of a probabilistic model.

- The question now is: what do the model parameters \( c, \alpha, \tau, \) and \( \mu \) really represent?

---

**Duration of a state**

Let us look at a single state with probability \( p \) of transitioning outside

\[ 1 - p \]

What is the probability of staying in that state for a duration \( D \)?
Duration of a state

\[
P[D=1] = \rho \\
P[D=2] = (1 - \rho) \rho \\
P[D=3] = (1 - \rho)^2 \rho \\
\vdots \\
P[D=D] = (1 - \rho)^{D-1} \rho
\]

i.e. geometric distribution

This is a disadvantage of a Markov Chain: it imposes a geometric distribution on the duration in a state, which might not be true.

There are ways around that… (see Durbin p. 69)

Mean duration

What is the mean duration?

\[
E[D] = 1/\rho
\]

Therefore...

\[\begin{align*}
\Delta = \mu_i \lambda_i & \quad \text{\( \lambda_i \) = mean length of ungapped regions} \\
\mu_i / \lambda_i & \quad \log \lambda_i \\
1 - \delta = \mu_g / \lambda_g & \quad \text{\( \lambda_g \) = mean length of gap} \\
\mu_g / \lambda_g & \quad \log \lambda_g \\
1 = \mu_r / \lambda_r & \quad \text{\( \lambda_r \) = mean length of related sequences} \\
(-\delta) & \\
\mu = \mu_l \lambda_l & \quad \text{\( \lambda_l \) = mean length of a sequence} \\
(-\delta)
\end{align*}\]