CS 466
Introduction to Bioinformatics

Instructor: Jian Peng
Teaching Assistant: Wesley Qian
Generate a sequence from a HMM

Hidden

\[ p(s(i)|s(i-1)) \quad p(s(i+1)|s(i)) \]

Observed

\[ p(x(i-1)|s(i-1)) \quad p(x(i)|s(i)) \quad p(x(i+1)|s(i+1)) \]
Generate a sequence from a HMM

Hidden: temperature
Observed: number of ice creams

HHHHHHHCCCCCCCCCHHHHHH
3323332111111233332

\[ B_1 = \begin{bmatrix} P(1 | \text{HOT}) \\ P(2 | \text{HOT}) \\ P(3 | \text{HOT}) \end{bmatrix} = \begin{bmatrix} .2 \\ .4 \\ .4 \end{bmatrix} \]

\[ B_2 = \begin{bmatrix} P(1 | \text{COLD}) \\ P(2 | \text{COLD}) \\ P(3 | \text{COLD}) \end{bmatrix} = \begin{bmatrix} .5 \\ .4 \\ .1 \end{bmatrix} \]
Hidden Markov Models: Applications

Speech recognition

[Yamato et al. CVPR 1992]: Tennis plays

Action recognition
Motif Finding

Problem:
**Find frequent motifs with length L in a sequence dataset**

```
ATCGCGCGCAGGTAAGGCGCGCGGCGCGGAAATCGDTATGCGCGCGCGCC
CAGGTAAGGTATTATGCGAGAGCATGCGCTATT
GTAGGCTGATGTTGGGGGCAAGGTAAGT
CGAGGAGTGCATG
CTAGGGAAACCGCGCGCGCGCGCGCGGAT
```

Assumption: the motifs are very similar to each other but look very different from the rest part of sequences
Motif: a first approximation

Assumption 1: lengths of motifs are fixed to L

Assumption 2: states on different positions on the sequence are independently distributed

\[ p_i(A) = \frac{N_i(A)}{N_i(A) + \sum_{j=1}^{L} N_i(j)} \]

\[ p(x) = \prod_{i=1}^{L} p_i(x(i)) \]
Motif: (Hidden) Markov models

Assumption 1: lengths of motifs are fixed to \( L \)

Assumption 2: future letters depend only on the present letter

\[
p_i(A|G) = \frac{N_{i-1,i}(G, A)}{N_{i-1}(G)}
\]

\[
p(x) = p_1(x(1)) \prod_{i=2}^{L} p_i(x(i)|x(i-1))
\]
Motif Finding

Problem:
We don’t know the exact locations of motifs in the sequence dataset

ATCGCGCGGCAGGTAAGGTATCGCGCGCC
CAGGTAAGGTATTATGCGAGACGATGTGCTATT
GTAGGCTGATGTGGGGGG
AAGGTAAGGTCGAGGAGTGCATG
CTAGGGAAACCGCGCGCGCGCGAT
CTAGGGAAACCGCGCGCGCGCGAT

Assumption: the motifs are very similar to each other but look very different from the rest part of sequences
Hidden state space

start

null

end

-3 -2 -1 1 2 3 4 5 6
Hidden Markov Model (HMM)

- Start
- Null (A = 0.25, C = 0.25, G = 0.25, T = 0.25)
- End

Transition Probabilities:
- Start to Null: 0.9
- Null to Null: 0.99
- Null to End: 0.02
- End to Null: 0.05
- Internal States:
  - A = 0.05, C = 0, G = 0.95, T = 0
  - A = 0.4, C = 0.1, G = 0.1, T = 0.4
How to build HMMs?
Computational problems in HMMs
Hidden Markov Models

\[ Q = q_1 q_2 \ldots q_N \] a set of \( N \) states

\[ A = a_{11} a_{12} \ldots a_{n1} \ldots a_{nn} \] a transition probability matrix \( A \), each \( a_{ij} \) representing the probability of moving from state \( i \) to state \( j \), s.t. \( \sum_{j=1}^{n} a_{ij} = 1 \) \( \forall i \)

\[ O = o_1 o_2 \ldots o_T \] a sequence of \( T \) observations, each one drawn from a vocabulary \( V = v_1, v_2, \ldots, v_V \)

\[ B = b_i(o_t) \] a sequence of observation likelihoods, also called emission probabilities, each expressing the probability of an observation \( o_t \) being generated from a state \( i \)

\( q_0, q_F \) a special start state and end (final) state that are not associated with observations, together with transition probabilities \( a_{01} a_{02} \ldots a_{0n} \) out of the start state and \( a_{1F} a_{2F} \ldots a_{nF} \) into the end state
Conditional Probability of Observations

\[ P(O|Q) = \prod_{i=1}^{T} P(o_i|q_i) \]

Example:

\[ P(3 \ 1 \ 3|\text{hot hot cold}) = P(3|\text{hot}) \times P(1|\text{hot}) \times P(3|\text{cold}) \]
Joint and marginal probabilities

Joint:

\[ P(O, Q) = P(O|Q) \times P(Q) = \prod_{i=1}^{n} P(o_i|q_i) \times \prod_{i=1}^{n} P(q_i|q_{i-1}) \]

\[ P(3 \ 1 \ 3, \text{hot hot cold}) = P(\text{hot}|\text{start}) \times P(\text{hot}|\text{hot}) \times P(\text{cold}|\text{hot}) \times P(3|\text{hot}) \times P(1|\text{hot}) \times P(3|\text{cold}) \]

Marginal:

\[ P(O) = \sum_{Q} P(O, Q) = \sum_{Q} P(O|Q)P(Q) \]

\[ P(3 \ 1 \ 3) = P(3 \ 1 \ 3, \text{cold cold cold}) + P(3 \ 1 \ 3, \text{cold cold hot}) + P(3 \ 1 \ 3, \text{hot hot cold}) + \ldots \]
How to compute the probability of observations

**Computing Likelihood:** Given an HMM $\lambda = (A, B)$ and an observation sequence $O$, determine the likelihood $P(O|\lambda)$.

$$P(O) = \sum_Q P(O, Q) = \sum_Q P(O|Q)P(Q)$$

For an HMM with $N$ hidden states and an observation sequence of $T$ observations, there are $N^T$ possible hidden sequences. For real tasks, where $N$ and $T$ are both large, $N^T$ is a very large number, so we cannot compute the total observation likelihood by computing a separate observation likelihood for each hidden state sequence and then summing them.

$$\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j|\lambda)$$ represents the probability of being in state $j$ after seeing the first $t$ observations, given the automaton $\lambda$. The value of each cell $\alpha_t(j)$ is computed by summing over the probabilities of every path that could lead us to this cell.

Here, $q_t = j$ means “the $t$th state in the sequence of states is state $j$”. We compute this probability $\alpha_t(j)$ by summing over the extensions of all the paths that lead to the current cell. For a given state $q_j$ at time $t$, the value $\alpha_t(j)$ is computed as

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i)a_{ij}b_j(o_t)$$
Forward algorithm

\[ \alpha_{t-1}(i) \quad \text{the previous forward path probability from the previous time step} \]
\[ a_{ij} \quad \text{the transition probability from previous state } q_i \text{ to current state } q_j \]
\[ b_j(o_t) \quad \text{the state observation likelihood of the observation symbol } o_t \text{ given the current state } j \]

1. Initialization:

\[ \alpha_1(j) = a_{0j}b_j(o_1) \quad 1 \leq j \leq N \]

2. Recursion (since states 0 and F are non-emitting):

\[ \alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i)a_{ij}b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T \]

3. Termination:

\[ P(O|\lambda) = \alpha_T(q_F) = \sum_{i=1}^{N} \alpha_T(i)a_{iF} \]
Forward algorithm

\[
\alpha_t(j) = \sum_i \alpha_{t-1}(i) a_{ij} b_j(o_t)
\]
Forward algorithm
Decoding: finding the most probable states

**Decoding**: Given as input an HMM $\lambda = (A, B)$ and a sequence of observations $O = o_1, o_2, \ldots, o_T$, find the most probable sequence of states $Q = q_1 q_2 q_3 \ldots q_T$.

Similar to the forward algorithm, we can define the following value:

$$v_t(j) = \max_{q_0, q_1, \ldots, q_{t-1}} P(q_0, q_1 \ldots q_{t-1}, o_1, o_2 \ldots o_t, q_t = j | \lambda)$$

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i) \ a_{ij} \ b_j(o_t)$$

- $v_{t-1}(i)$: the previous Viterbi path probability from the previous time step
- $a_{ij}$: the transition probability from previous state $q_i$ to current state $q_j$
- $b_j(o_t)$: the state observation likelihood of the observation symbol $o_t$ given the current state $j$
1. Initialization:

\[ v_1(j) = a_{0j}b_j(o_1) \quad 1 \leq j \leq N \]
\[ b_{t_1}(j) = 0 \]

2. Recursion (recall that states 0 and \(q_F\) are non-emitting):

\[ v_t(j) = \max_{i=1}^{N} v_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T \]
\[ b_{t}(j) = \arg\max_{i=1}^{N} v_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T \]

3. Termination:

The best score: \(P^* = v_T(q_F) = \max_{i=1}^{N} v_T(i) * a_{iF}\)

The start of backtrace: \(q_T^* = b_{T}(q_F) = \arg\max_{i=1}^{N} v_T(i) * a_{iF}\)