Hidden Markov Models

Lecture 6, Thursday April 17, 2003

1. When the true underlying states are known

Given \( x = x_1 \ldots x_N \) for which the true \( \pi = \pi_1 \ldots \pi_N \) is known,

**Define:**

\[
A_{kl} = \text{# times } k \rightarrow l \text{ transition occurs in } \pi \\
E_k(b) = \text{# times state } k \text{ in } \pi \text{ emits } b \text{ in } x
\]

We can show that the maximum likelihood parameters \((a, e)\) are:

\[
a_{kl} = \frac{A_{kl}}{\Sigma_i A_{ki}} \quad e_k(b) = \frac{E_k(b)}{\Sigma_c E_k(c)}
\]

Lecture 7, Tuesday April 22, 2003

2. When not – The Baum-Welch Algorithm

**Initialization:**

Pick the best-guess for model parameters (or arbitrary)

**Iteration:**

- Forward
- Backward
- Calculate \( A_{kl}, E_k(b) \)
- Calculate new model parameters \( a_{kl}, e_k(b) \)
- Calculate new log-likelihood \( P(x | \theta) \)

GUARANTEED TO BE HIGHER BY EXPECTATION-MAXIMIZATION

Until \( P(x | \theta) \) does not change much

Lecture 7, Tuesday April 22, 2003

Alternative: Viterbi Training

**Initialization:**

Same

**Iteration:**

- Perform Viterbi, to find \( \pi^* \)
- Calculate \( A_{kl}, E_k(b) \) according to \( \pi^* \) + pseudocounts
- Calculate the new parameters \( a_{kl}, e_k(b) \)

Until convergence

**Notes:**

- Convergence is guaranteed – Why?
- Does not maximize \( P(x | \theta) \)
- In general, worse performance than Baum-Welch
- Convenient – when interested in Viterbi parsing, no need to implement additional procedures (Forward, Backward)!!

Lecture 7, Tuesday April 22, 2003

Variants of HMMs

Lecture 6, Thursday April 17, 2003
Higher-order HMMs

The Genetic Code
3 nucleotides make 1 amino acid
Statistical dependencies in triplets

Question:
Recognize protein-coding segments with a HMM

One way to model protein-coding regions

P(x_{i-3}, x_{i-2}, x_{i-1} | x_i, x_{i+1}, x_{i+2})

Every state of the HMM emits 3 nucleotides

Transition probabilities:
Probability of one triplet, given previous triplet: P(π_i | π_{i-1})

Emission probabilities:
P(x_{i-3}, x_{i-2}, x_{i-1} | π_i) = 1/0
P(x_{i-1}, x_{i-2}, x_{i-3} | π_{i-1}) = 1/0

A more elegant way

Every state of the HMM emits 1 nucleotide
Transition probabilities:
Probability of one triplet, given previous 3 triplets: P(π_i | π_{i-1}, π_{i-2}, π_{i-3})

Emission probabilities:
P(x_i | π_i)

Algorithms extend with small modifications

Modeling the Duration of States

Length distribution of region X:
E[l_X] = 1/(1-p)

- Exponential distribution, with mean 1/(1-p)

This is a significant disadvantage of HMMs

Several solutions exist for modeling different length distributions

Solution 1: Chain several states

Disadvantage: Still very inflexible
l_X = C + exponential with mean 1/(1-p)

Solution 2: Negative binomial distribution

P(l_X = n) = \sum_{k=1}^{n} \binom{n-1}{k-1} p^k (1-p)^{n-k}

Algorithms extend with small modifications
Solution 3: Duration modeling

Upon entering a state:
1. Choose duration d, according to probability distribution
2. Generate d letters according to emission probs
3. Take a transition to next state according to transition probs

Disadvantage: Increase in complexity:
Time: $O(D^2)$
Space: $O(D)$
Where $D =$ maximum duration of state

---

Connection Between Alignment and HMMs

A state model for alignment

Alignments correspond 1-to-1 with sequences of states M, I, J

Let’s score the transitions

Alignments correspond 1-to-1 with sequences of states M, I, J

How do we find optimal alignment according to this model?

Dynamic Programming:
- $M(i, j)$: Optimal alignment of $x_1...x_i$ to $y_1...y_j$ ending in $M$
- $I(i, j)$: Optimal alignment of $x_1...x_i$ to $y_1...y_j$ ending in $I$
- $J(i, j)$: Optimal alignment of $x_1...x_i$ to $y_1...y_j$ ending in $J$

The score is additive, therefore we can apply DP recurrence formulas

Needleman Wunsch with affine gaps – state version

Initialization:
- $M(0, 0) = 0$
- $M(i, 0) = M(0, j) = -\infty$, for $i, j > 0$

Iteration:
- $M(i, j) = s(x_i, y_j) + \max(M(i-1, j-1), I(i-1, j), J(i-1, j))$
- $I(i, j) = \max(I(i-1, j-1), M(i, j-1) + e, J(i-1, j) + d)$
- $J(i, j) = \max(I(i, j-1) + e, J(i-1, j) + d, M(i-1, j) + e)$

Termination:
- Optimal alignment given by $\max\{M(m, n), I(m, n), J(m, n)\}$
Probabilistic interpretation of an alignment

An alignment is a hypothesis that the two sequences are related by evolution

**Goal:**

- Produce the most likely alignment
- Assert the likelihood that the sequences are indeed related

A Pair HMM for alignments

A Pair HMM for not aligned sequences

To compare ALIGNMENT vs. RANDOM hypothesis

Every pair of letters contributes:

- \((1 - 2 \delta - \tau) P(x_i, y_j)\) when matched
- \(\epsilon P(x_i) P(y_j)\) when gapped
- \((1 - \eta)^2 P(x_i) P(y_j)\) in random model

Focus on comparison of

- \(P(x_i, y_j)\) vs. \(P(x_i) P(y_j)\)

To compare ALIGNMENT vs. RANDOM hypothesis

We will divide alignment score by the random score, and take logarithms

Let

- \(s(x_i, y_j) = \log \frac{P(x_i, y_j)}{P(x_i) P(y_j)} \frac{1 - 2 \delta - \tau}{1 - \eta}\)
- \(d = -\log \frac{\epsilon}{(1 - \eta)(1 - \eta)(1 - 2 \delta - \tau) P(x_i)}\)
- \(e = -\log \frac{\eta}{(1 - \eta) P(b)}\)
The meaning of alignment scores

Because $\delta$, $\epsilon$, are small, and $\eta$, $\tau$ are very small,

$$s(x_i, y_j) = \log \frac{P(x_i, y_j)}{P(x_i) P(y_j)} \approx \log \frac{1}{1 - 2\delta - \tau} + \log \frac{1}{1 - \eta}$$

$$d = -\log \frac{1}{1 - \eta} \approx -\log \delta$$

$$e = -\log \frac{1}{1 - \eta} \approx -\log \epsilon$$

Lecture 7, Tuesday April 22, 2003

The meaning of alignment scores

The Viterbi algorithm for Pair HMMs corresponds exactly to the Needleman-Wunsch algorithm with affine gaps

However, now we need to score alignment with parameters that add up to probability distributions

$\delta$: 1 mean arrival time of next gap
$\epsilon$: 1 mean length of next gap
$\tau$: affine gaps decouple arrival time with length
$\eta$: 1 mean length of conserved segments (set to $\sim0$)
$\eta$: 1 mean length of sequences of interest (set to $\sim0$)

Lecture 7, Tuesday April 22, 2003

Substitution matrices

A more meaningful way to assign match/mismatch scores

For protein sequences, different substitutions have dramatically different frequencies!

BLOSUM matrices:

1. Start from BLOCKS database (curated, gap-free alignments)
2. Cluster sequences according to % identity
3. For a given L% identity, calculate $A_{ab}$: # of aligned pos a-b
4. Estimate

$$P(a) = \frac{\sum b A_{ab}}{\sum cd A_{cd}}; \quad P(a, b) = \frac{A_{ab}}{\sum cd A_{cd}}$$

Lecture 7, Tuesday April 22, 2003

BLOSUM matrices

The two are scaled differently

Lecture 7, Tuesday April 22, 2003

Example:

Say DNA regions between human and mouse have average conservation of 50% Then $P(A,A) = P(C,C) = P(G,G) = P(T,T) = 1/8$ (so they sum to 1/2) $P(A,C) = P(A,G) = \ldots = P(T,G) = 1/24$ (24 mismatches, sum to 1/2) $P(A) = P(C) = P(G) = P(T) = 1/4$

$$\log \left( \frac{1/8}{1/4 \times 1/4} \right) = \log 2 = 1, \text{ for match}$$

$$\log \left( \frac{1/24}{1/4 \times 1/4} \right) = \log 16/24 = -0.585$$

Note: 0.585 / 1.585 = 37.5

Why? 37.5% is between the 50% conservation model, and the random 25% conservation model !

Lecture 7, Tuesday April 22, 2003