Part Two:
Decoding: The Viterbi Algorithm

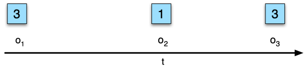
Zihao Zhu, Gangze Li, Yufeng Xie

Tsinghua University

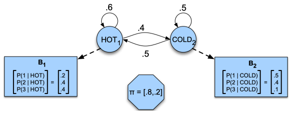
November 5, 2019

Decoding

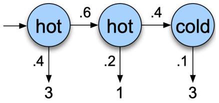
Decoding: Given as input an HMM $\lambda = (A, B)$ and a sequence of observations $O = o_1, o_2, \ldots, o_T$, find the most probable sequence of states $Q = q_1 q_2 q_3 \ldots q_T$.



(a) observations



(b) $\lambda = (A, B)$



(c) hidden weather sequence

Figure: In the ice-cream domain, given a sequence of ice-cream observations 3 1 3 and an HMM, the task of the decoder is to find the best hidden weather sequence (H H H).

Problem

- One possibility
  - For each hidden state sequence Q
    - HHH, HHC, HCH, etc.
  - Compute $P(O|Q)$
  - Pick the highest one
- ⚠ likelihood: N^T
  a set of N states, a sequence of T observations
- Instead
  - Viterbi algorithm

Problem

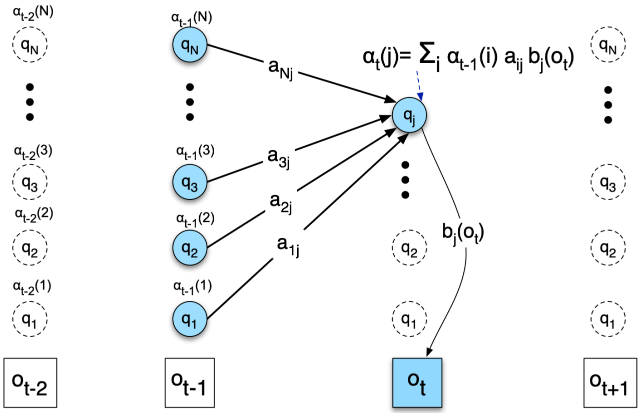


Figure: exponentially large number of state sequences! [1]

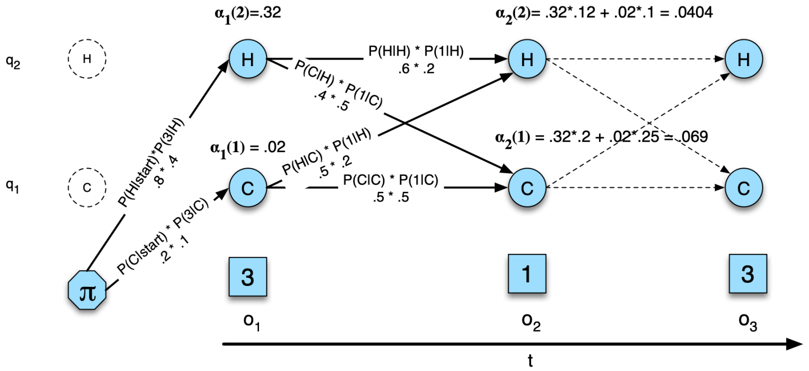
[1] example from [JM19]

Viterbi algorithm

- The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states.
- We want to compute the joint probability of the observation sequence together with the best state sequence.
  - $v_t(j) = P(q_0, q_1, \ldots, q_{t-1}, o_1, o_2, \ldots, o_t, q_t = j|\lambda)$
  - $\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j|\lambda)$

The forward trellis

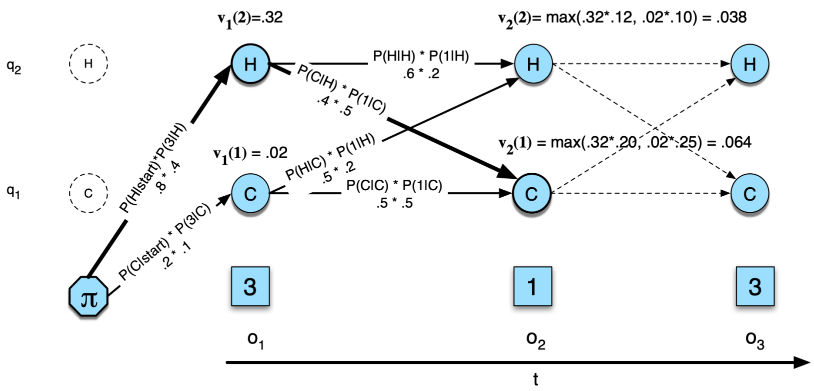
compute the total observation likelihood for the ice-cream events 3 1 3



$$\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j|\lambda)$$

Viterbi algorithm

the computation of $v_t(j)$ for two states at two time steps



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

Viterbi algorithm

- The computation in each cell follows
  $v_t(j) = \max_{1 \leq i \leq N-1} v_{t-1}(i) a_{ij} b_j(o_t)$
- The resulting probability expressed in each cell is
  $v_t(j) = P(q_0, q_1, \ldots, q_{t-1}, o_1, o_2, \ldots, o_t, q_t = j|\lambda)$

the Viterbi backtrace

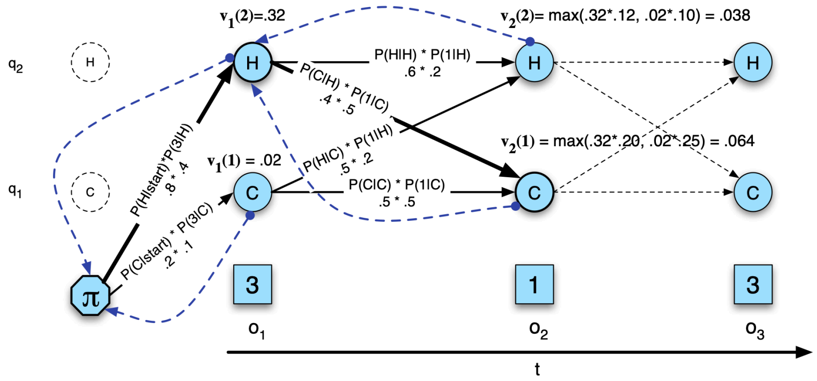


Figure: The Viterbi trellis for computing the best path through the hidden state space for the ice-cream eating events 3 1 3.

Viterbi recursion

- Initialization

$$\begin{aligned} v_1(j) &= \pi_j b_j(o_1) & 1 \leq j \leq N \\ bt_1(j) &= 0 & 1 \leq j \leq N \end{aligned}$$

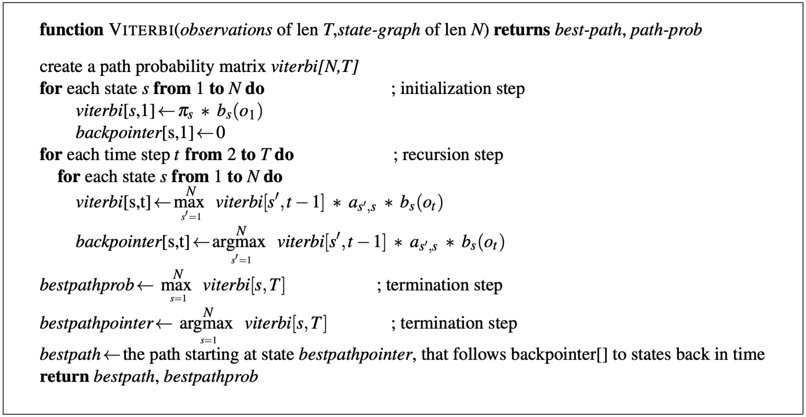
- Recursion

$$\begin{aligned} v_t(j) &= \max_{i=1}^{N} v_{t-1}(i) a_{ij} b_j(o_t)\,; \quad 1 \leq j \leq N, 1 < t \leq T \\ bt_t(j) &= \operatorname{argmax}_{i=1}^{N} v_{t-1}(i) a_{ij} b_j(o_t)\,; \quad 1 \leq j \leq N, 1 < t \leq T \end{aligned}$$

- Termination

The best score: $P* = \max_{i=1}^{N} v_T(i)$
The start of backtrace: $q_{T^*} = \operatorname{argmax}_{i=1}^{N} v_T(i)$

Pseudocode



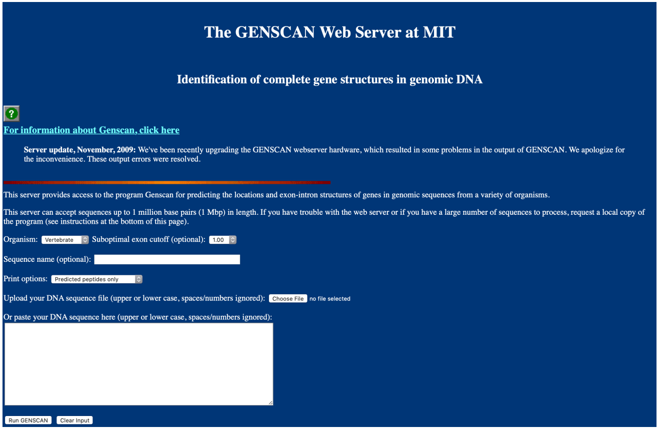
```
function VITERBI(observations of len $T$, state-graph of len $N$) returns best-path, path-prob

create a path probability matrix viterbi[N,T]
for each state $s$ from 1 to $N$ do                    ; initialization step
      $viterbi[s,1] \leftarrow \pi_s * b_s(o_1)$
      $backpointer[s,1] \leftarrow 0$
for each time step $t$ from 2 to $T$ do                ; recursion step
   for each state $s$ from 1 to $N$ do
      $viterbi[s,t] \leftarrow \max_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$
      $backpointer[s,t] \leftarrow \operatorname{argmax}_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$
$bestpathprob \leftarrow \max_{s=1}^{N} viterbi[s,T]$              ; termination step
$bestpathpointer \leftarrow \operatorname{argmax}_{s=1}^{N} viterbi[s,T]$        ; termination step
$bestpath \leftarrow$ the path starting at state bestpathpointer, that follows backpointer[] to states back in time
return bestpath, bestpathprob
```

Three fundamental problems [Rab89]

- **Problem 1 (Likelihood)** Given an HMM $= (A, B)$ and an observation sequence O, determine the likelihood $P(O|\lambda)$.
- **Problem 2 (Decoding)** Given an observation sequence O and an HMM $= (A, B)$, discover the best hidden state sequence Q.
- **Problem 3 (Learning)** Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

Application[2]



The GENSCAN Web Server at MIT

Identification of complete gene structures in genomic DNA

For information about Genscan, click here

Server update, November, 2009: We've been recently upgrading the GENSCAN webserver hardware, which resulted in some problems in the output of GENSCAN. We apologize for the inconvenience. These output errors were resolved.

This server provides access to the program Genscan for predicting the locations and exon-intron structures of genes in genomic sequences from a variety of organisms.

This server can accept sequences up to 1 million base pairs (1 Mbp) in length. If you have trouble with the web server or if you have a large number of sequences to process, request a local copy of the program (see instructions at the bottom of this page).

Organism: Vertebrate Suboptimal exon cutoff (optional): 1.00

Sequence name (optional):

Print options: Predicted peptides only

Upload your DNA sequence file (upper or lower case, spaces/numbers ignored): Choose File no file selected

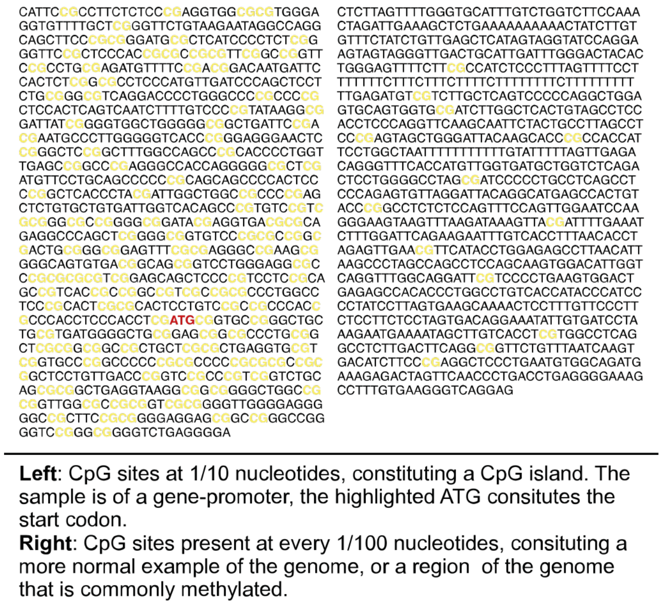
Or paste your DNA sequence here (upper or lower case, spaces/numbers ignored):

[2]from the link

GENSCAN

GENSCAN was developed by Chris Burge in the research group of Samuel Karlin, Department of Mathematics, Stanford University [BK97, SSK98].
Burge and Karlin in their gene finding computer program GENSCAN used a three-periodic (inhomogeneous) fifth order Markov chain to model coding regions of DNA sequences [BE06].

CpG island[3]



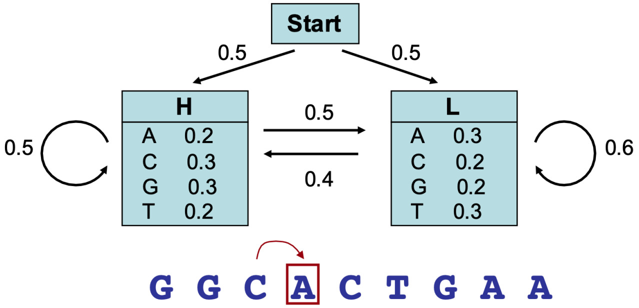
```
CATTCCGCCTTCTCTCCCGAGGTGGCGCGTGGGA
GGTGTTTTGCTCGGGTTCTGTAAGAATAGGCCAGG
CAGCTTCCCGCGGGATGCGCTCATCCCCTCTCGG
GGTTCCGCTCCCACCGCGCCGCGTTCGGCCGGTT
CCGCCTGCGAGATGTTTTCCGACGGACAATGATTC
CACTCTCGGCGCCTCCCATGTTGATCCCAGCTCCT
CTGCGGGCGTCAGGACCCCTGGGCCCCGCCCCG
CTCCACTCAGTCAATCTTTTGTCCCCGTATAAGGCG
GATTATCGGGGTGGCTGGGGGCGGCTGATTCCGA
CGAATGCCCTTGGGGGTCACCCGGGAGGGAACTC
CGGGCTCCGGCTTTGGCCAGCCCGCACCCCTGGT
TGAGCCGGCCCGAGGGCCACCAGGGGGCGCTCG
ATGTTCCTGCAGCCCCCCGCAGCAGCCCCACTCC
CCGGCTCACCCTACGATTGGCTGGCCGCCCCGAG
CTCTGTGCTGTGATTGGTCACAGCCCGTGTCCGTC
GCGGGCGCCGGGGCGGATACGAGGTGACGCGCA
GAGGCCCAGCTCGGGGCGGTGTCCCGCGCCGGC
GACTGCGGGCGGAGTTTCGCGAGGGCCGAAGCG
GGGCAGTGTGACGGCAGCGGTCCTGGGAGGCGC
CCGCGCGCGTCGGAGCAGCTCCCCGTCCTCCGCA
GCCGTCACCGCCCGGCCGTCGCCGCGCCCTGGCC
TCCCGCACTCGCGCACTCCTGTCCGCCCGCCCACC
GCCCACCTCCCACCTCGATGCGGTGCCGGGCTGC
TGCGTGATGGGGCTGCGGAGCGGCGCCCTGCGG
CTCGCGGCGGCCGCTGCTCGCGCTGAGGTGCGT
CGGTGCCCGGCCCCCCGCGCCCCCGCGCGCCGC
GGCTCCTGTTGACCCGGTCCGCCCGTCGGTCTGC
AGCGCGGCTGAGGTAAGGCGGCGGGGCTGGCCG
CGGTTGGCGCCGCGGTCGCGGGGTTGGGGAGGG
GGCCGCTTCCGCGGGGAGGAGCGGCCGGGCCGG
GGTCCGGGCGGGGTCTGAGGGGA
```

```
CTCTTAGTTTTGGGTGCATTTGTCTGGTCTTCCAAA
CTAGATTGAAAGCTCTGAAAAAAAAAACTATCTTGT
GTTTCTATCTGTTGAGCTCATAGTAGGTATCCAGGA
AGTAGTAGGGTTGACTGCATTGATTTGGGACTACAC
TGGGAGTTTTCTTCGCCATCTCCCTTTAGTTTTCCT
TTTTTTCTTTCTTTCTTTTCTTTTTTTTTCTTTTTTTTT
TTGAGATGTCGTCTTGCTCAGTCCCCCAGGCTGGA
GTGCAGTGGTGCGATCTTGGCTCACTGTAGCCTCC
ACCTCCCAGGTTCAAGCAATTCTACTGCCTTAGCCT
CCCGAGTAGCTGGGATTACAAGCACCCGCCACCAT
TCCTGGCTAATTTTTTTTTTTTGTATTTTTAGTTGAGA
CAGGGTTTCACCATGTTGGTGATGCTGGTCTCAGA
CTCCTGGGGCCTAGCGATCCCCCTGCCTCAGCCT
CCCAGAGTGTTAGGATTACAGGCATGAGCCACTGT
ACCCGGCCTCTCTCCAGTTTCCAGTTGGAATCCAA
GGGAAGTAAGTTTAAGATAAAGTTACGATTTTGAAAT
CTTTGGATTCAGAAGAATTTGTCACCTTTAACACCT
AGAGTTGAACGTTCATACCTGGAGAGCCTTAACATT
AAGCCCTAGCCAGCCTCCAGCAAGTGGACATTGGT
CAGGTTTGGCAGGATTCGTCCCCTGAAGTGGACT
GAGAGCCACACCCTGGCCTGTCACCATACCCATCC
CCTATCCTTAGTGAAGCAAAACTCCTTTGTTCCCTT
CTCCTTCTCCTAGTGACAGGAAATATTGTGATCCTA
AAGAATGAAAATAGCTTGTCACCTCGTGGCCTCAG
GCCTCTTGACTTCAGGCGGTTCTGTTTAATCAAGT
GACATCTTCCCGAGGCTCCCTGAATGTGGCAGATG
AAAGAGACTAGTTCAACCCTGACCTGAGGGGAAAG
CCTTTGTGAAGGGTCAGGAG
```

Left: CpG sites at 1/10 nucleotides, constituting a CpG island. The sample is of a gene-promoter, the highlighted ATG consitutes the start codon.
Right: CpG sites present at every 1/100 nucleotides, consituting a more normal example of the genome, or a region of the genome that is commonly methylated.

[3]from the link

Toy Example [4]

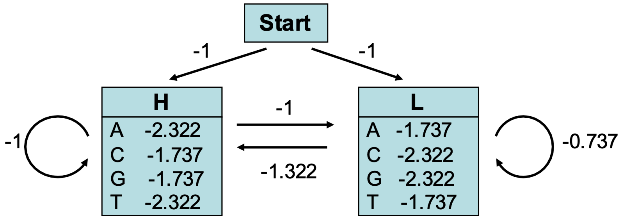


This model is composed of 2 states, H (high GC content) and L (low GC content). We can for example consider that state H characterizes coding DNA while L characterizes non-coding DNA.

$$p_H(A,4) = e_H(A)\max\left(p_L(C,3)p_{LH}, p_H(C,3)p_{HH}\right)$$

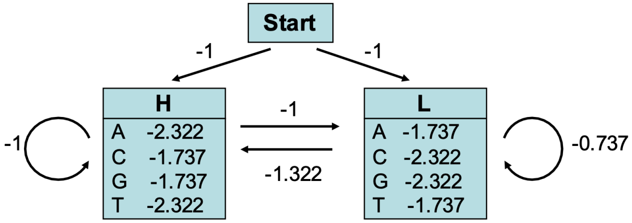
[4]example from the link

Toy Example



We used here $log2(p)$, and compute sums instead of products.

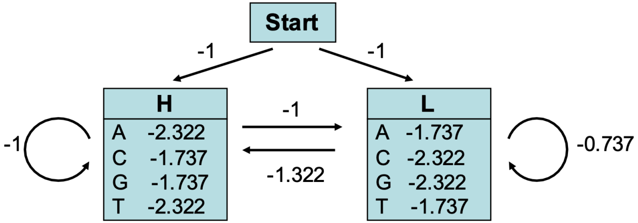
Toy Example



GGCACTGAA

- Probability (in log_2) that G at the first position was emitted by state H, $p_H(G,1) = -1 - 1.737 = -2.737$
- Probability (in log_2) that G at the first position was emitted by state L, $p_L(G,1) = -1 - 2.322 = -3.322$

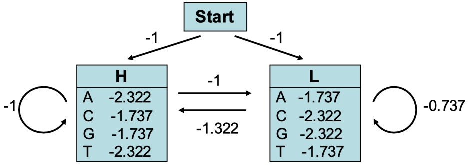
Toy Example



GGCACTGAA

- Probability (in log_2) that G at the 2nd position was emitted by state H,
$$p_H(G,2) = -1.737 + \max\left(p_H(G,1) + p_{HH}, p_L(G,1) + p_{LH}\right)$$
$$= -1.737 + \max(-2.737 - 1, -3.322 - 1.322)$$
$$= -5.474 \text{(obtained from } p_H(\mathrm{G},1))$$
- Probability (in log_2) that G at the 2nd position was emitted by state L,
$$p_L(G,2) = -2.322 + \max\left(p_H(G,1) + p_{HL}, p_L(G,1) + p_{LL}\right)$$
$$= -2.322 + \max(-2.737 - 1.322, -3.322 - 0.737)$$
$$= -6.059 \text{(obtained from } p_H(\mathrm{G},1))$$

Toy Example

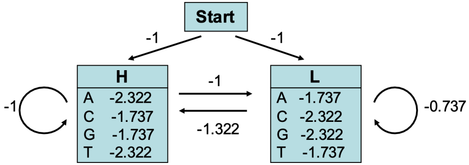


GGCACTGAA

	G	G	C	A	C	T	G	A	A
H	-2.73	-5.47	-8.21	-11.53	-14.01	...			-25.65
L	-3.32	-6.06	-8.79	-10.94	-14.01	...			-24.49

We then compute iteratively the probabilities $p_H(i,x)$ and $p_L(i,x)$ that nucleotide i at position x was emitted by state H or L, respectively. The highest probability obtained for the nucleotide at the last position is the probability of the most probable path. This path can be retrieved by back-tracking.

Toy Example



GGCACTGAA

back-tracking
(= finding the path which corresponds to the highest probability, -24.49)

	G	G	C	A	C	T	G	A	A
H	**-2.73**	**-5.47**	**-8.21**	-11.53	-14.01	...			-25.65
L	-3.32	-6.06	-8.79	**-10.94**	**-14.01**	...			**-24.49**

The most probable path is: HHHLLLLLL

References I

- Mark Borodovsky and Svetlana Ekisheva, *Problems and solutions in biological sequence analysis*, Cambridge University Press, 2006.
- Chris Burge and Samuel Karlin, *Prediction of complete gene structures in human genomic dna*, Journal of molecular biology **268** (1997), no. 1, 78–94.
- Dan Jurafsky and James H. Martin, *Speech and language processing (3rd ed. draft)*, 2019.
- Lawrence R Rabiner, *A tutorial on hidden markov models and selected applications in speech recognition*, Proceedings of the IEEE **77** (1989), no. 2, 257–286.
- SL Salzberg, DB Searls, and S Kasif, *Modeling dependencies in pre-mrna splicing signals*, Computational methods in molecular biology **32** (1998), 129.

References II