COL776

Report of Assignment 2(b)

DEVANSH DALAL (2012CS10224)

Note: All the code is mainly implemented in Python.

Q1 Finding CPGs islands in the sequence

Commands : python 2b.py <path to the testing file>

Where path to the training data is assumed to be *../gene_data/* .

Cpg islands in testing file found by inference

Results	Start	end
1	4075	5678
2	6343	6755
3	7340	7544
4	8161	8418
5	31893	32279
6	41518	41883
7	47627	48242
8	49232	49400
9	53230	54102

Viterbi Algorithm:

$$V_{1,k} = P(x_1|y_1 = k) * \pi_k$$

$$V_{1,k} = argmax_{x \in S} P(x_t|y_t = k) * T_{x,k} * V_{t-1,x}$$

        To avoid overflows, I have used log of the probabilities instead of probabilities in the computations of probabilities of hidden states for each index i. And also no state cannot be assigned zero probability as log(0) is not defined so we assign a very small number (10^{-30}) if probability is zero. In this approach the log-probability of the most likely sequence will be

$$\log(prob) = argmax_{x \in S} V_{L-1,x}$$

For test data the log of probability of most likely state was found to be

$$\log(prob\ of\ most\ likly\ sequence) = -104885.387535$$

Time complexity:

      The transition, emission and initial probabilities are computed as mentioned in the assignment and Viterbi algorithm is implemented for calculating the most likely states of the hidden sequence.

$$complexity : O(n * |S|^2)$$

where n is the length of the sequence and S is the set of all possible states. Since we have |S|=8 for all indexes of the sequence. Therefore final complexity will be.

$$complexity : O(n)$$

Q2 Connection between Max-product VE and Viterbi

      The max-product VE and Viterbi both necessarily calculates the same result i.e. most-probable cumulative assignment of variables over the given Markov network. The max-product VE is more general as compared to the Viterbi algorithm applied here on the linear chain. But they calculates the same inference.

References

1. http://www.cse.iitd.ac.in/~parags/teaching/col776/assignments/ass1-b/ass2-b.pdf
2. https://en.wikipedia.org/wiki/Viterbi_algorithm