Gene discovery using Hidden Markov Model

DS202: Algorithmic Foundations of Big Data Biology

Presented by:
Dyotana Das - 19223
Vidushi Dwivedi - 19606
(30-03-2022)

Topics to be covered

1. Introduction to HMM
2. The three central problems of HMM:

a. Evaluation (Forward and backward algo)

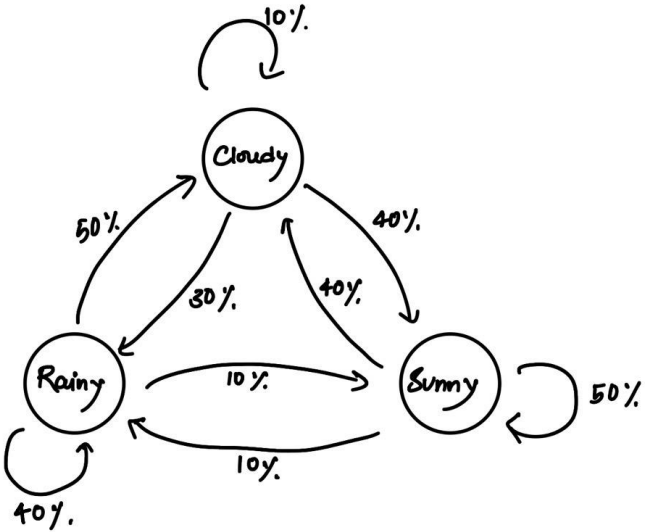
b. Decoding (Viterbi algo)

c. Learning (Baum Welch algo)

3. HMM being applied for Gene discovery

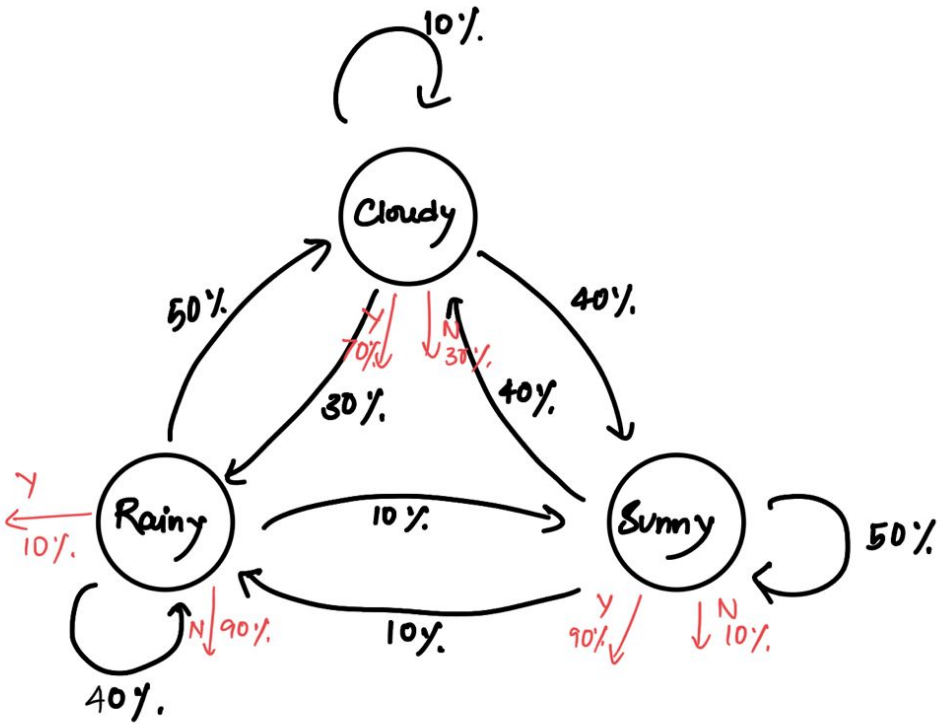
Markov Chain

- Stochastic model
- Key property: The probability of each even depends only on the previous one (Markov Assumption)



Hidden Markov Model

- Emissions associated with each state.
- Only the emissions are observed (state sequence hidden)
- **Additional property**: each emission depends only on one state (the one it is associated with)
- Characterized by $(A,B,\pi) = \lambda$



Given sets of observations we can ask:

1. How probable is a given observation sequence?

2. What is the best underlying state sequence for a given observation sequence?

3. For a set of observations, can we determine the HMM parameters best suiting to it?

Three central problems of HMM

1. How probable is a given observation sequence? **EVALUATION**

2. What is the best underlying state sequence for a given observation sequence? **DECODING**

3. For a set of observations, can we determine the HMM parameters best suiting to it? **LEARNING**

Problem 1: EVALUATION

- Finding probability of given observation sequence [$P(O/\lambda)$ where O is observation sequence and λ is the HMM]
- $P(O/\lambda) = \sum_q P(O,q/\lambda)$
- Computationally extensive [$O(TN^T)$ for T time instances and N states]
- Thus we move to Forward algo and Backward algo : $O(N^2T)$

Forward Method

- Define a forward variable $\alpha_t(i)$ [for partial sequence upto time instance t, probability of stopping at ith state at time instance t]
-

Initialize:

$$\alpha_1(i) = \pi_i b_i(o_1)$$

Recursion:

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}] b_j(o_{t+1})$$

Final Result:

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

Working Example of Forward algo

Discrete HMM - Leela is doing a term project on using HMMs as a generative model. She uses a two state discrete HMM. She assumes a simple model with self transition probabilities $a_{11} = 0.8$ and $a_{22} = 0.8$ and initial probability of $\pi_1 = 0.6$. Further, the HMM emits only binary symbols with $b_1(1) = 0$ and $b_2(1) = 1$. Let o_t indicate the symbol emitted at time t. In one of the experiments, she observes $o_3 = 0, o_4 = 0, o_5 = 1$. Find the probability of this observation sequence ?

Backward Method

- Define a forward variable $\beta_t(i)$ [for partial sequence from t upto time instance T, probability of stopping at ith state at time instance t]
- Initialization:

$$\beta_T(j) = 1$$

Recursion:

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(o_{t+1})$$

Final Result:

$$p(O|\lambda) = \sum_{i=1}^{N} \beta_1(i) \qquad t = T-1,\ldots,1$$

Problem 2: DECODING

- Finding the best state sequence for a given observation sequence
- Use Viterbi algorithm to reduce time complexity (keeps best state sequence at each instance)
- Define variable δ

$$\delta_t(i) = \max_q P(q_1, q_2, \ldots, q_t = i, o_1, o_2, \ldots o_t \mid \lambda)$$

$\delta_t(i)$ – the probability of the most probable path ending in state $q_t=i$

Viterbi (continued)

Recurrent property:

$$\delta_{t+1}(j) = \max_i(\delta_t(i)a_{ij})b_j(o_{t+1})$$

Algorithm:

1. Initialise:

$$\delta_1(i) = \pi_i b_i(o_1) \qquad 1 \le i \le N$$

$$\psi_1(i) = 0$$

Viterbi (continued)

2. Recursion:

$$\delta_t(j) = \max_{1 \le i \le N} (\delta_{t-1}(i) a_{ij}) b_j(o_t)$$

$$\psi_t(j) = \arg\max_{1 \le i \le N} (\delta_{t-1}(i) a_{ij}) \qquad 2 \le t \le T, 1 \le j \le N$$

3. Termination:

$$P^* = \max_{1 \le i \le N} \delta_T(i)$$

$$q_T^* = \arg\max_{1 \le i \le N} \delta_T(i)$$

Viterbi (continued)

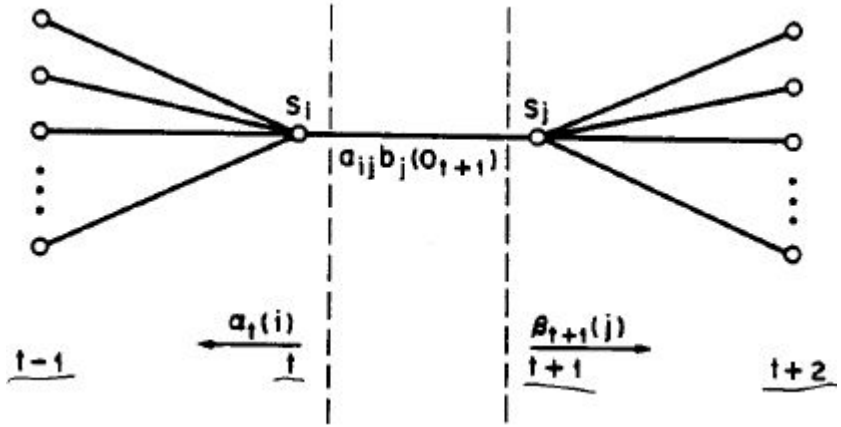
4. Backtracking state sequence

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \qquad t + T - 1, T - 2, \ldots, 1$$

Problem 3: LEARNING

- Method to adjust the model parameters (A,B,π)
- Maximizes the probability of the observation sequence (Baum Welch Algorithm)
- EM algorithm - an iterative method to find (local) maximum likelihood estimates of parameters in statistical models.

Defining some variables:



$$\xi_t(i,j) = \frac{P(q_t = i,\ q_{t+1} = j,\ \mathbf{O} \mid \lambda)}{P(\mathbf{O} \mid \lambda)}$$
$$= \frac{\alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}{P(\mathbf{O}|\lambda)}$$
$$= \frac{\alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}.$$

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j).$$

$\sum_{t=1}^{T-1} \gamma_t(i)$ = expected number of transitions from state i in $\mathbf{O}$

$\sum_{t=1}^{T-1} \xi_t(i,j)$ = expected number of transitions from state i to state j in $\mathbf{O}$.

Estimation of parameters:

$$\bar{\pi}_i = \text{expected frequency (number of times) in state } i \text{ at time } (t = 1) = \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_j(k) = \frac{\text{expected number of times in state } j \text{ and observing symbol } \mathbf{v}_k}{\text{expected number of times in state } j} = \frac{\sum_{\substack{t=1 \\ \text{s.t. } \mathbf{o}_t = \mathbf{v}_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}.$$

Derivation of the re-estimation formulas from Q function

$$Q(\lambda', \lambda) = \sum_{\mathbf{q}} P(\mathbf{O}, \mathbf{q}|\lambda') \log P(\mathbf{O}, \mathbf{q}|\lambda)$$

$$Q(\lambda', \lambda) = Q_{\pi}(\lambda', \pi) + \sum_{i=1}^{N} Q_{a_i}(\lambda', \mathbf{a}_i) + \sum_{i=1}^{N} Q_{b_i}(\lambda', \mathbf{b}_i)$$

$$P(\mathbf{O}, \mathbf{q}|\lambda) = \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(\mathbf{o}_t)$$

$$\bar{\pi}_i = \frac{\alpha_0(i)\beta_0(i)}{\sum_{j=1}^{N} \alpha_T(j)} = \gamma_0(i)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T} \alpha_{t-1}(i)\, a_{ij} b_j(\mathbf{o}_t)\beta_t(j)}{\sum_{t=1}^{T} \alpha_{t-1}(i)\beta_{t-1}(i)} = \frac{\sum_{t=1}^{T} \xi_{t-1}(i,j)}{\sum_{t=1}^{T} \gamma_{t-1}(i)}$$

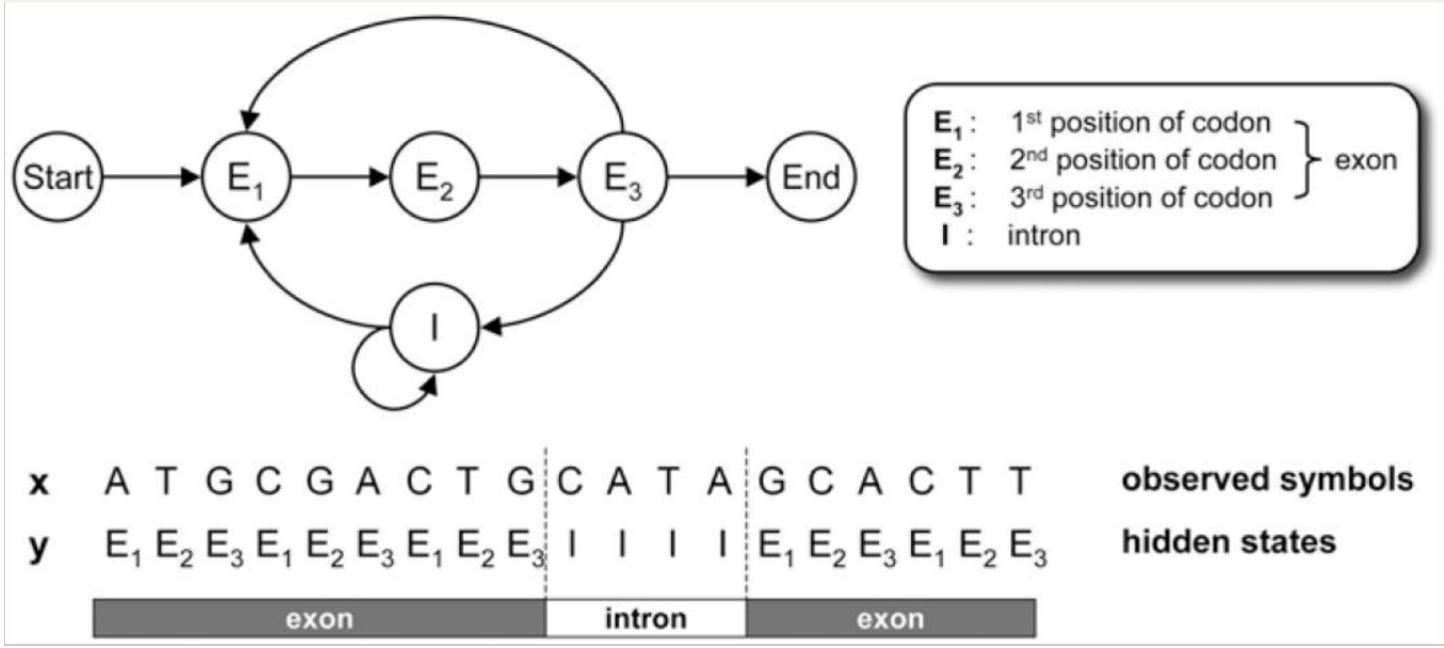
$$\bar{b}_i(k) = \frac{\sum_{t=1}^{T} \alpha_t(i)\beta_t(i)\delta(\mathbf{o}_t, \mathbf{v}_k)}{\sum_{t=1}^{T} \alpha_t(i)\beta_t(i)} = \frac{\sum_{t=1 \text{ s.t.} \mathbf{o}_t = \mathbf{v}_k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}$$

Initial estimates of HMM parameters:

- Random or uniform initial estimates for A and π
- For B parameter

    1. Manual segmentation of the observation and averaging
    2. Maximum likelihood estimation of the observation and averaging
    3. Segmental K means segmentation

Simple HMM model for Eukaryotic genes



HMM that models protein-coding genes in eukaryotes

Q1) Can we find out whether this DNA sequence is a coding gene or not?

➔ observation probability of x based on the given HMM that models coding genes

Q2) Can we predict the locations of the exons and introns in the given sequence?

➔ predict the state sequence y in the HMM that best describes x.Now,predict the locations of the exons and introns in y.

HMMs provide a formal probabilistic framework for analyzing biological sequences.

Summary of Work Done

- Understood and implemented Forward, Backward, Viterbi and Baum Welch Algorithms
- Tested the implementation on audio data
- Understood the implementation of HMM for Gene Finding

Will move ahead with experimenting the algos on Genomic data

Thank you!