AI CS362

LAB 6

Team Alpha:

- Sparsh Agrawal
- Rushikesh Jadhav
- Vivek Kumar
- Rathod Yashwanth

Objective

To implement Expectation Maximization routine for learning parameters of a Hidden Markov Model, to be able to use the EM framework for deriving algorithms for problems with hidden or partial information.

Problem Statement

1. Read through the reference carefully. Implement routines for learning the parameters of HMM given in section 7. In section 8, “A not-so-simple example”, an interesting exercise is carried out. Perform a similar experiment on “War and Peace” by Leo Tolstoy.
2. Ten bent (biased) coins are placed in a box with unknown bias values. A coin is randomly picked from the box and tossed 100 times. A file containing results of five hundred such instances is presented in tabular form with 1 indicating head and 0 indicating tail. Find out the unknown bias values. (2020_ten_bent_coins.csv) To help you, a sample code for two bent coin problem along with data is made available in the work folder: two_bent_coins.csv and embentcoinsol.m
3. A point set with real values is given in 2020_em_clustering.csv. Considering that there are two clusters, use EM to group together points belonging to the same cluster. Try and argue that k-means is an EM algorithm.

Explanation

HMM

A Markov model is a method for randomly changing systems that possess the Markov property. This means that, at any given time, the next state is only dependent on the current state and is independent of anything in the past.

In a Hidden Markov Model, the state of the system is hidden and can only be inferred from the observed outputs, hence the name "hidden". The HMM is defined by two things:

1. The state transition probabilities, which define the probability of transitioning from one state to another.
2. The observation probabilities, which define the probability of observing a specific output given a specific state.

The HMM is used to model systems in which the state of the system changes over time and the state can only be indirectly observed through the output. The goal of HMM is to estimate the underlying state sequence given a sequence of observations. This is typically done using algorithms such as the Viterbi Algorithm, the Forward-Backward Algorithm, or the Baum-Welch Algorithm.

Notations

- T = length of the observation sequence
- N = number of states in the model
- M = number of observation symbols
- A = state transition probabilities
- B = observation probability matrix
- π = initial state distribution
- $O = (O_0, O_1, \dots , O_{T-1})$ = observation sequence

Main problems we want to solve by HMM

1. **Given the model λ = (A, B, π) and a sequence of observations O, find P(O | λ). Here, we want to determine a score for the observed sequence O with respect to the given model λ.**

   We can use forward algorithm to find the forward pass(α) for the given model.

   $P(O|\lambda) = \Sigma \ \alpha_{T-1}(i)$

2. **Given λ = (A, B, π) and an observation sequence O, find an optimal state sequence for the underlying Markov process. In other words, we want to uncover the hidden part of the Hidden Markov Model.**

   Forward-Backward algorithm is used to find the optimal state sequence given the model.

   Here, we want to maximize $P(X_t \mid O_0 O_1 \ldots\ldots O_{T-1} , \lambda)$

   By Bayes theorem,

   $P(X_t \mid O_0 O_1 \ldots\ldots O_{T-1} , \lambda) = P(O_{t+1} \ldots\ldots O_{T-1} \mid X_t , \lambda)\ P(O_0 O_1 \ldots\ldots O_t X_t \mid , \lambda) / P(O_0 O_1 \ldots\ldots O_{T-1} \mid \lambda)$

   $\gamma_t (i) = \alpha_t (i)\ \beta_t (i) / \boldsymbol{\Sigma}\ \ \alpha_{T-1} (i)$

   $\alpha_t (i) = [\boldsymbol{\Sigma}\ \ \alpha_{t-1} (j)\ a_{ij}]\ b_j(O_t)$

   $\beta_t (i) = \boldsymbol{\Sigma}\ a_{ij}\ b_j (O_{t+1})\ .\ \beta_{t+1}(j)$

3. **Given an observation sequence O and the dimensions N and M, find the model λ = (A, B, π) that maximizes the probability of O. This can be viewed as training a model to best fit the observed data.**

   In this question we want to find the λ such that P(O | λ) is maximized. Here we do not have the model.So we are going to use the Baum-Welch algorithm. It is going to use Expectation Maximization (EM).

   The E-step:

   - Make an initial guess for a , b and pie.
   - Compute the forward and backward probabilities for a given model and reconstruct the hidden structure.

   The M-step:

   - Using that hidden structure re-estimate the model parameters.
   - Repeat this until convergence

HMM Scaling

HMM scaling is necessary because the probabilities in an HMM are often represented in log space for numerical stability and computational efficiency. $\alpha_t(i)$ tends to 0 exponentially as T increases. Therefore, any attempt to implement the formulae as given above will inevitably result in underflow. The solution to this underflow problem is to scale the numbers.

$$\alpha_t\text{‘}(i) = \alpha_t(i) / \boldsymbol{\Sigma}\ \alpha_t(j)$$

$$\log[P(O \mid \lambda)] = -\Sigma \log c_j$$

Re-estimate Parameters

$\gamma_t(i) = P(x_t = i \mid O , \lambda)$

$\gamma_t(i , j) = P(x_t = i ,x_t = j \mid O, \lambda) = \alpha_t(i)\, a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j) / P(O \mid \lambda)$

$a'_{ij} = \Sigma_{t=1}^{T-1} \gamma_t(i , j) \;/\; \Sigma_{t=1}^{T-1} \gamma_t(i)$

$b'_i(O_k) = \Sigma_{t=1}^{T-1} \gamma_t(i)\, 1(Ot = k) \;/\; \Sigma_{t=1}^{T-1} \gamma_t(i)$

Expectation Maximization Algorithm:

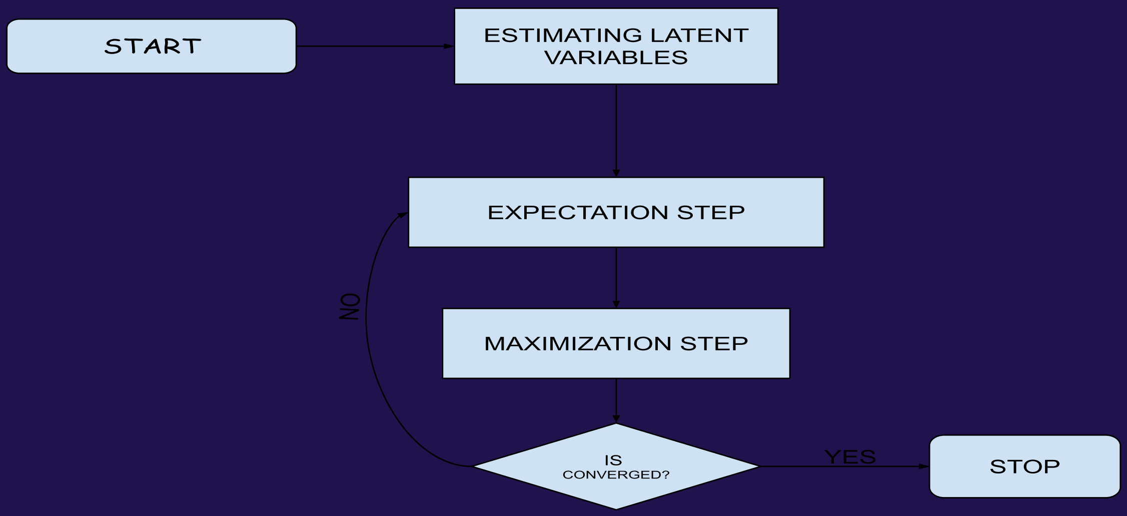
Expectation Maximization is an iterative optimization method to estimate some unknown parameters Θ, given measurement data U.

Algorithm:
Step 1) Estimate the missing or latent variables

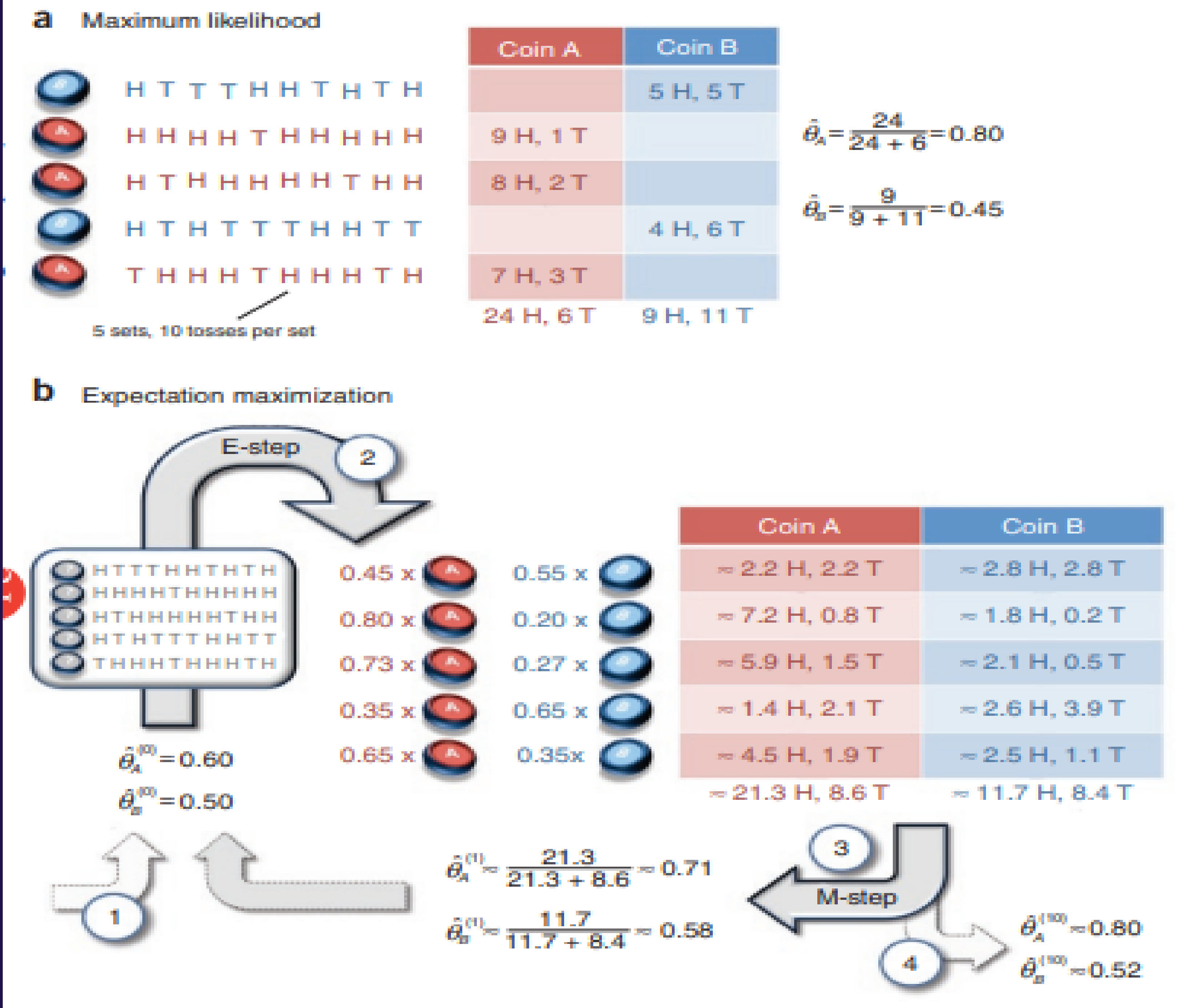
Step 2) Expectation step(E): Guess the missing data with the help of observed data.

Step 3) Maximization step(M): Use the data generated in the Expectation step to update the values of parameters.

Step 4) Repeat steps 2 and 3 until the values are converging.



Let’s understand with an example:



Coin A	Coin B
	5 H, 5 T
9 H, 1 T	
8 H, 2 T	
	4 H, 6 T
7 H, 3 T	
24 H, 6 T	9 H, 11 T

$$\hat{\theta}_A = \frac{24}{24+6} = 0.80$$

$$\hat{\theta}_B = \frac{9}{9+11} = 0.45$$

Coin A	Coin B
≈ 2.2 H, 2.2 T	≈ 2.8 H, 2.8 T
≈ 7.2 H, 0.8 T	≈ 1.8 H, 0.2 T
≈ 5.9 H, 1.5 T	≈ 2.1 H, 0.5 T
≈ 1.4 H, 2.1 T	≈ 2.6 H, 3.9 T
≈ 4.5 H, 1.9 T	≈ 2.5 H, 1.1 T
≈ 21.3 H, 8.6 T	≈ 11.7 H, 8.4 T

$$\hat{\theta}_A^{(0)} = 0.60$$

$$\hat{\theta}_B^{(0)} = 0.50$$

$$\hat{\theta}_A^{(1)} \approx \frac{21.3}{21.3+8.6} \approx 0.71$$

$$\hat{\theta}_B^{(1)} \approx \frac{11.7}{11.7+8.4} \approx 0.58$$

$$\hat{\theta}_A^{(10)} \approx 0.80$$

$$\hat{\theta}_B^{(10)} \approx 0.52$$

K-Means Algorithm

- K-Means algorithm is an iterative algorithm that tries to partition the dataset into K pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group.
- It tries to make the intra-cluster data points as similar as possible while also keeping the

  clusters as different (far) as possible.
- It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster centroid (arithmetic mean of all the data points that belong to that cluster) 1s at the minimum.

Algorithm : k-means algorithm

1: Specify the number k of clusters to assign.

2: Randomly initialize k centroids.

3: Repeat

4: Expectation: Assign each point to its closest centroid.

5: Maximization: Compute the new centroid (mean) of each cluster.

6: Until The centroid positions do not change.

References

- A Revealing Introduction to Hidden Markov Models, Mark Stamp, 2018
- What is the expectation maximization algorithm? Chuong B Do and Serafim Batzoglou, Nature Biotechnology, Vol 26, Num 8, August 2008
- https://github.com/nlok5923/CS302