Lab Assignment 6
Implementing Expectation Maximization for Learning Parameters of a Hidden Markov Model

Khushi Saxena
Computer Science Engineering
202151078@iiitvadodara.ac.in

Makwana Harsh Maheshkumar
Computer Science Engineering
202151082@iiitvadodara.ac.in

Jagruti Piprade
Computer Science Engineering
202151067@iiitvadodara.ac.in

Nitin Gautam
Computer Science Engineering
202151101@iiitvadodara.ac.in

Python notebook code for this assignment can be found here.

***Abstract*—This report explores the implementation of the Expectation Maximization (EM) routine for learning parameters of a Hidden Markov Model (HMM). The learning objectives include understanding the EM framework and its application to problems with hidden or partial information. We detail the problem statement, which involves implementing routines for learning HMM parameters and performing experiments based on a provided reference. Additionally, we conduct experiments involving ten bent coins to determine unknown bias values and perform clustering using EM on a dataset with real values. Through rigorous experimentation and analysis, we aim to gain insights into the effectiveness and versatility of the EM framework in handling problems with hidden information.**

I. Problem Statement

- 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.

- 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 problems along with data is made available in the work folder: two-bent-coins.csv and embentcoinsol.m

- 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.

II. Introduction

In this report, we explore the implementation of the Expectation Maximization (EM) routine for learning parameters of a Hidden Markov Model (HMM). The learning objectives include understanding the EM framework and its application to problems with hidden or partial information. We begin by discussing the problem statement and the tasks to be performed based on the reference provided. We then delve into the implementation of routines for learning the parameters of the HMM, as outlined in section 7 of the reference. Subsequently, we undertake an interesting experiment based on "War and Peace" by Leo Tolstoy, following a similar approach outlined in section 8 of the reference.

III. Implementation of EM for HMM

A Markov process is a random process that is indexed by time. It is a stochastic model that describes a sequence of possible events. The probability of each event depends only on the state attained in the previous event.

Hidden Markov Models (HMMs) are probabilistic models used for modeling sequential data, where the underlying system is assumed to be a Markov process with unobservable (hidden) states.

We detail the implementation of the EM routine for learning the parameters of the HMM. This involves iterative steps of expectation and maximization to update the parameters until convergence is achieved.

IV. Experiment on "War and Peace"

Following the experiment outlined in section 8 of the reference, we conducted a similar analysis of "War and Peace" by Leo Tolstoy. We aim to extract hidden information using the EM framework and derive insights from the text.

V. Experiment with Ten Bent Coins

Expectation Maximization (EM) is an iterative algorithm used to find maximum likelihood or maximum posterior estimates of parameters in statistical models where some variables are unobserved or missing. It is particularly useful in cases where the model depends on latent variables, which are variables that are not directly observed but are inferred from observed data.

We analyze the dataset containing the results of coin tosses for ten bent coins and determine the unknown bias values using EM. We discuss the methodology, implementation details, and the results obtained.

A. How the EM algorithm works

- **Expectation Step (E-step):** In this step, you make an educated guess about the missing information based on the data you have. You calculate the probabilities or likelihoods of different scenarios given what you know.
- **Maximization Step (M-step):** Now that you have your educated guess from the E-step, you use it to update your knowledge about the missing information. You adjust your parameters or guesses to better fit the data.
- **Iteration:** Steps 2 and 3 are repeated iteratively until the algorithm converges, meaning that the parameter estimates stop changing significantly between iterations or until a maximum number of iterations is reached.

Experiment 1: We see which coin is flipped.

coin	flips	# coin 1 heads	# coin 2 heads
B	HTTTHHTHTH	0	5
A	HHHHTHHHHH	9	0
A	HTHHHHHTHH	8	0
B	HTHTTTHHTT	0	4
A	THHHTHHHTH	7	0

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

This means that if we toss coin A 80% it will come up with Heads.

Experiment 2: We don't see which coin is flipped

coin	flips	# coin 1 heads	# coin 2 heads
?	HTTTHHTHTH	?	?
?	HHHHTHHHHH	?	?
?	HTHHHHHTHH	?	?
?	HTHTTTHHTT	?	?
?	THHHTHHHTH	?	?

flips	probability it was coin C1	probability it was coin C2	# heads attributed to C1	# heads attributed to C2
HTTTHHTHTH	0.45	0.55	5 * 0.45 = 2.25	5 * 0.55 = 2.75
HHHHTHHHHH	0.8	0.2	9 * 0.8 = 7.2	9 * 0.2 = 1.8
HTHHHHHTHH	0.73	0.27	8* 0.73 = 5.84	2 * 0.27 = 2.16
HTHTTTHHTT	0.35	0.65	4 * 0.35 = 1.4	4 * 0.65 = 2.6
THHHTHHHTH	0.65	0.35	7 * 0.65 = 4.55	7 * 0.35 = 2.45

VI. Clustering using EM

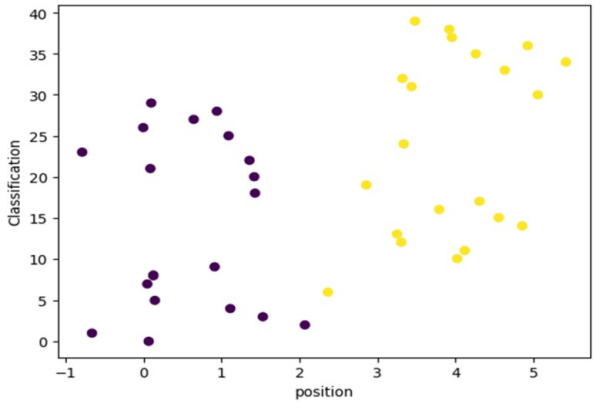
K-means is another popular algorithm used for clustering data points into groups or clusters. It's a simpler and more

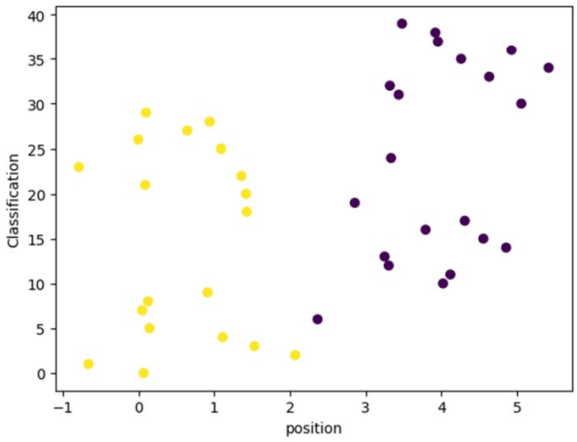
computationally efficient method compared to the EM algorithm, particularly when dealing with large datasets.

A. How K-means works:

- **Initialization:** We choose K initial cluster centroids randomly from the data points. K represents the number of clusters we want to identify.
- **Assignment Step:** We assign each data point to the nearest centroid. This is typically done by calculating the Euclidean distance (or other distance metrics) between each data point and each centroid and assigning the data point to the nearest centroid.
- **Update Step:** Recalculate the centroids of the clusters by taking the mean of all data points assigned to each centroid. This moves the centroids to the center of their respective clusters.
- **Iteration:** Repeating the assignment and update steps iteratively until convergence criteria are met. Typically, convergence occurs when the centroids no longer change significantly between iterations or when a maximum number of iterations is reached.

Using the provided dataset (2020-em-clustering.csv), we perform clustering with EM to group together points belonging to the same cluster. We compare the results with k-means clustering and discuss the relationship between the two algorithms.





VII. Conclusion

In conclusion, we have successfully implemented the EM routine for learning the parameters of an HMM and applied it to various experiments as outlined in the problem statement. Through rigorous experimentation and analysis, we have gained insights into the effectiveness and versatility of the EM framework in handling problems with hidden or partial information. Future work may involve further exploration of advanced EM techniques and their application to real-world datasets.