Problem #4

Problem statement :

4. Estimate parameters for a simple model using the EM algorithm: What are the chances of heads and tails from two unlabeled coins? To figure it out you will implement an EM algorithm that estimates the probability of heads for two coins using data from an API. For more details on the EM Algorithm, read this short article that uses our task as the example. One hint: don't do everything from scratch. You are allowed to use libraries like scipy or scikitlearn.
    4.1. Create python code that can access the Heads/Tails api
        4.1.1. The heads/tails api returns 20 "coin flips" (1s represent Heads and 0s represent tails). For each api call, it will return 20 "coin flips" from the same coin, but it will randomly select from two different coins with each having different probability of heads.
        4.1.2. The API is located at the following address: https://24zl01u3ff.execute-api.us-west-1.amazonaws.com/beta
        4.1.3. Parse the returned json object for the list of 20 1s or 0s in the "body" section of the json
    4.2. Write python code that creates an EM Algorithm that will take a set of 30 coin flip draws (a draw is the returned 20 flips from the api) and uses the series of 30 draws to estimate the theta parameters for the two coins (theta is the likelihood of heads which are 1s in this example).
    4.3. Create a writeup explaining the development of your algorithm. Make sure to give your estimates of the thetas for the two coins. Beyond a minimal bar of technical competence, you will be assessed largely on the clarity of your write-up.

Methodology:

Language used: Python
Library used: pandas, numpy, sklearn, nltk

Little About EM Algorithm

It is not an "algorithm", it is a Framework.
It has a loop of two phases.

- Estimation (using Maximum likelihood)
- Modification (using Expectation maximization)

Here we are given a set of coin flips (20 coin flips and 30 set/draws) but we don't know which coin is sampled for each draw.

Task : Figure out the likelihood of heads of each coin !
Data: Sequence of coin flips: X = [1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0]

We can assume any thetas value initially. The Idea is to use an EM algorithm over time to eventually get the best thetas value.

We would be running the following two steps in a loop till we reach the convergence.
Loop
Let Θ_A= 0.6 ((likelihood of head for coin A)
Let Θ_B= 0.5 ((likelihood of head for coin B)

Step 1. Estimation (using Maximum likelihood)

- Sampling the coin with 20 coin-flips, say X = $\{x_{1,} x_{2,} \ldots x_{30,}\}$ where $x_{1,}$ means sampling a coin with 20 flips and X as data of 20 coin-flips over 30 draws (here API is giving the coin-flips over 30 draws). Let's say these samples of 20 coin-flips are in 30 Rows.

- Calculate the probability of coin A and coin B given each Row (total 30 Rows with 20 coin-flips)
  P (coin A | Row) = P (coin A , Row) / p(Row) ; using conditional probability
  P (coin B | Row) = P (coin B , Row) / p(Row) ; using conditional probability

- Calculate total number of heads and tails for each coin against each Row/draw.
  expected number of heads considering the coin is A = P (coin A | Row) * number of heads from the coin-flips

  expected number of tails considering the coin is A = P (coin A | Row) * number of tails from the coin-flips

  expected number of heads considering the coin is B = P (coin B | Row) * number of heads from the coin-flips

  expected number of tails considering the coin is B = P (coin B | Row) * number of tails from the coin-flips

- Sum the total number of expected heads and tails for coin A and coin B separately.

Step 2. Maximization update : Update the thetas value from the expected sampling result.

Θ_A^{new} = expected number of heads using coin A / total number of flips using coin A

Θ_B^{new} = expected number of heads using coin B / total number of flips using coin B

repeat till the convergence

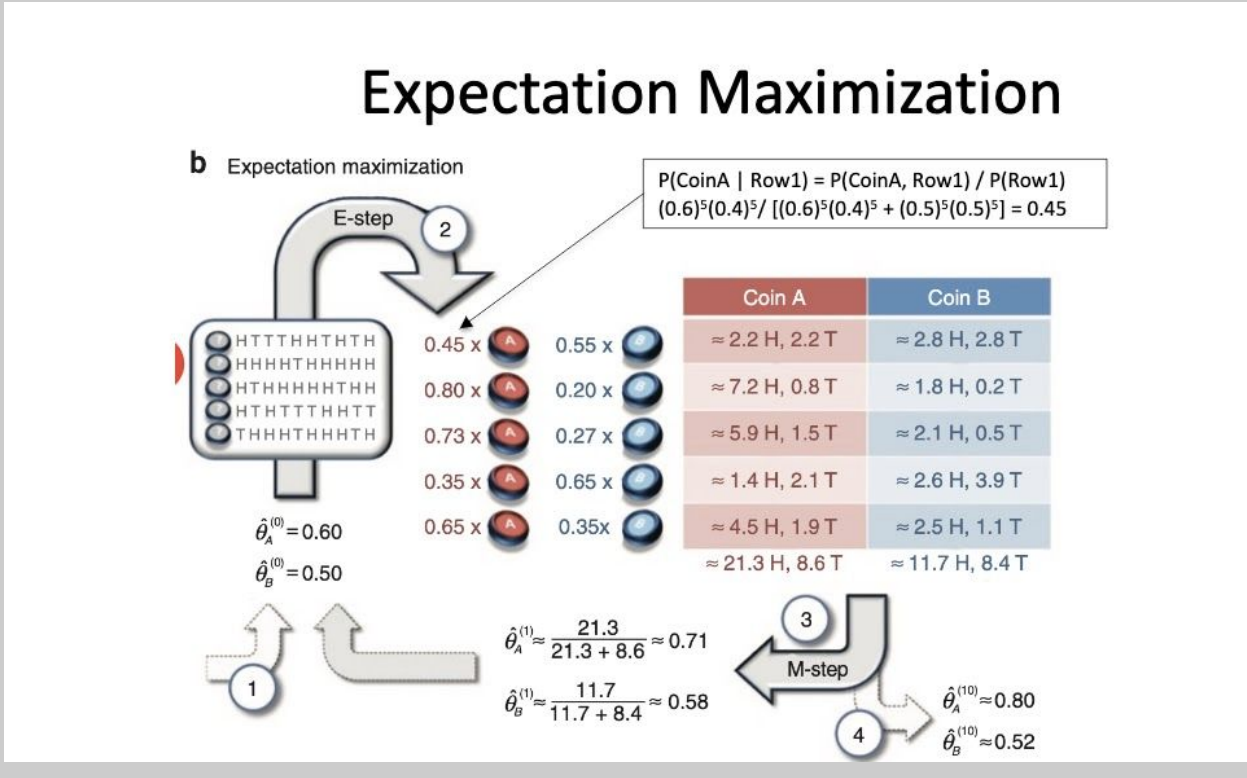

Fig: Showing working of EM algorithm (sampling with 10 coin-flips - in our case it is 20 coin-flips over 30 draws)

How do we know the convergence of thetas are achieved?

If the theta's values are not changing over iteration then we can conclude that we have reached convergence and these are the best likelihood values for coin A and coin B.
In the code, I am using combined log likelihood of coin A and coin B to conclude the same. If the new log likelihood value is approximately the same as the previous log

likelihood value then we are coming out of the loop to say that these are our converged thetas values.

Results

The thetas values will keep changing for the given problem because different runs will have different samplings coming from the given API.

Final thetas values for heads for both coin A and coin B respectively :
0.7422332797265275 0.3087849492510257
Chances of heads and tails respectively from coin A :
0.7422332797265275 0.25776672027347247
Chances of heads and tails respectively from coin B :
0.3087849492510257 0.6912150507489743

```
In [121]: # Driver code to test em algorithm for the given problem.

          samples = np.array(samples) # converting samples to a numpy array.

          # Initial random probability of (heads, tails) for coin A and coin B respectively.
          thetas = np.array([[0.6, 0.4], [0.5, 0.5]])
          thetas = em(samples, thetas)
          print("\n\nFinal thetas values for heads : ", thetas[0][0], thetas[1][0])
          print("Chances of heads and tails respectively from coin A :", thetas[0][0], thetas[0][1])
          print("Chances of heads and tails respectively from coin B :", thetas[1][0], thetas[1][1])

          New thetas values after iteration :  1 [0.69832347 0.30167653] [0.37307086 0.62692914]
          New thetas values after iteration :  2 [0.74208685 0.25791315] [0.31270459 0.68729541]
          New thetas values after iteration :  3 [0.74267246 0.25732754] [0.30929432 0.69070568]
          New thetas values after iteration :  4 [0.74234286 0.25765714] [0.30889302 0.69110698]
          New thetas values after iteration :  5 [0.74223328 0.25776672] [0.30878495 0.69121505]
          breaking when i =  4


          Final thetas values for heads :  0.7422332797265275 0.3087849492510257
          Chances of heads and tails respectively from coin A : 0.7422332797265275 0.25776672027347247
          Chances of heads and tails respectively from coin B : 0.3087849492510257 0.6912150507489743
```