Department of Mathematics

Unsurpervised Learning 1 Project

Written by

GUIDJIME ADINSI Ahouahounko Télesphore
BAH Mamadou Oury

Submitted to : Prof. **AMBROISE** Christophe

2022-2023

List of Figures

Contents

January 9, 2023

Introduction

In this project we will explore three problems. The first one is related to the page rank algorithm. The way this problem is stated in the project is a Markov Model with a certain transition matrix encoding the transition probabilities over websites. We will go through the details of the problem in section (**). The second problem, at its core is a Hidden Markov Model (HMM). Where there are hidden (unobservable) states and an observable process. The goal will be to infer the parameters of the model and the most likely sequence of hidden states given multiple sequences of observations. Finally, the last problem is that of community detection. At its core, we will focus on the Stochastic Block Model(SBM). Given block (communities), we will simulate from the model, and the goal will be to try and find a clustering algorithm.

1 Page Rank Algorithm:

Let's consider a directed graph G with K nodes. Every node corresponds to web page and the edges (directed) between nodes are hyperlinks pointing from one page to another. First we will simulated an oriented graph with $K = 8$ nodes using the Erdös-Renyi model. This model puts an edge between two nodes independently with probability $p = 0.4$. If we let $\mathbf{X}$ be the adjacency matrix where the element $X_{ij} = 1$ if there is an edge between node i and j and 0 otherwise. This model can mathematically be expressed as follows:

$$X_{ij} \sim \mathcal{B}ern(p) \; \forall \, i, j \in \{1, 2, ..., 8\}$$

In this setting we will allow loops on a node (i.e a website can point to itself). One could write their own code to simulate from the above model. However, we will use the package *igraph* from R to simulate the 8×8 matrix. Now since the Erdös-Renyi will generate a family of graphs we will set a seed a focus on only on member of the family the seed is set to 20222023. Bellow is the view of the graph.

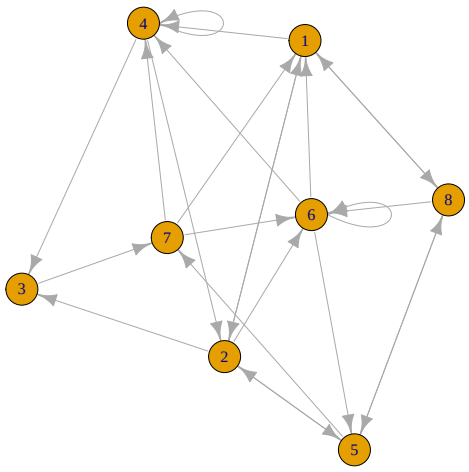

Figure 1: A directed Graph simulated using the Erdös-Renyi model

Now by assuming that a surfer will click at random on the available hyperlinks on the current page (links chosen with equi-probability) with probability $1-\epsilon$ and with probability ϵ the surfer goes to one of the 8 available websites. Note that from the description given above we can write the transition matrix associated to the Markov that describes a web surfing path followed by the random surfer on our graph. There are two processes associated with the description. First we begin by noticing that when the surfer is on any website out of the 8 available their is an *epsilon* probability that he decides to jump on any other of available website with equi-probability ($\frac{1}{8}$). Thus this part of be process can be described mathematical a matrix $\mathbf{M}$ where:

$$M_{ij}^1 = \frac{\epsilon}{K}$$

Moreover, we look at the second process associated with the surfing. This part is a bit subtle. First note that this side of the process is first chosen with probability $(1-\epsilon)$. The given that surfer chose this process he will then decide to move to the next website by choosing on the available links on that website (i.e which websites the current website points to) with equi-probability. We first notice that when we look in a given row of the adjacency matrix we can read the websites that this given row points to by the columns that have values 1. Thus equi-probability means dividing every element in that row the sum of the elements of that given row. Thus the transition matrix associated to this process is :

$$M_{ij}^2 = (1-\epsilon)\cdot\frac{X_{ij}}{\sum_j X_{ij}}$$

Finally since the random surfer chooses one or the other process, so the transition matrix is:

$$A_{ij} = M_{ij}^1 + M_{ij}^2 = (1-\epsilon)\cdot\frac{X_{ij}}{\sum_j X_{ij}} + \frac{\epsilon}{K},\ K = 8 \tag{1.0.1}$$

Thus from the adjacency matrix of the graph simulated in figure(1), we can compute the corresponding transition matrix that describes the surfer's path. We will fix ϵ to be 0.05. (The simulation and the codes will be in the rmarkdown file associated to the project).

1.1 The Stationary Distribution, and The MLE of The Transition Matrix

Given the transition matrix, we can ask ourselves what is the stationary distribution of Markov chains described by the transition matrix in 1.0.1. By assuming that the chain is **aperiodic** and **irreducible**, we will compute it's **stationary distribution**. The idea is to find long term distribution over the states, and we want to reach the stage where (ref Murphy page 597):

$$\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{A} \tag{1.1.1}$$

We note here that $\boldsymbol{\pi}$ is a row vector, and we can further note that 1.1.1 can be thought of finding the eigenvector $\mathbf{v}$ of $\mathbf{A}^\top$ associated with the eigenvalue 1. By making the observation that transposing equation(1.1.1) we get:

$$\mathbf{A}^\top\mathbf{v} = \mathbf{v}\,, \textit{where } \mathbf{v} = \boldsymbol{\pi}^\top \tag{1.1.2}$$

However, the eigenvector may not be a normalized probability, so we will need to normalize the vector(i.e divide each element of the vector by the sum of its elements.) to get a valid stationary distribution. After going through the computations in R and rounding the values to 3 decimal points we get:

$$\boldsymbol{\pi} = \big[0.138,\ 0.140,\ 0.097,\ 0.185,\ 0.095,\ 0.138,\ 0.128,\ 0.078\big]$$

Now we will simulate a sequence of 1000 successive clicks made by the random surfer according to the matrix simulated by eq (1.0.1). We will fixed the first website to be website number 2, websites will be labeled $\{1, 2, ..., 8\}$. Thus the sequence of clicks will look something like $\big[2\ \ 3\ \ 7\ \ 4\ \ 3\ \ 7\ \ 4\ \ 3.....\big]$. Meaning that the surfer started in website number 2 the moved to website 3 and website 7, and so on and so forth.
Given the above simulated data, let's estimate the transition matrix of the chain. We can begin by searching for the maximum likelihood estimator of the parameters of our chain. The derivation are given at A, and it implementation straight forward in R using the function **table**.

Since Above we have estimated the transition matrix $\hat{\mathbf{A}}$ given the simulation, let's use it to estimate the stationary distribution $\hat{\boldsymbol{\pi}}$ and compare it to $\boldsymbol{\pi}$. This is straight forward. We will use the same procedure as in Eq.(1.1.1), but we replace $\mathbf{A}$ by $\hat{\mathbf{A}}$ as in solving for $\mathbf{v}$ in:

$$\hat{\mathbf{A}}^\top\mathbf{v} = \mathbf{v}\,, \textit{where } \mathbf{v} = \hat{\boldsymbol{\pi}}^\top$$

After going trough the computation in R we get:

$$\hat{\boldsymbol{\pi}} = \big[0.129,\ 0.149,\ 0.107,\ 0.188,\ 0.087,\ 0.139,\ 0.138,\ 0.063\big]$$

Finally, as a way to compare the two distribution let's compute the $\mathcal{KL}$-divergence between the two distributions (i.e):

$$\mathcal{KL}(\boldsymbol{\pi}||\hat{\boldsymbol{\pi}}) = \sum_{k=1}^{K} \boldsymbol{\pi}_k \log(\frac{\boldsymbol{\pi}_k}{\hat{\boldsymbol{\pi}}_k})$$

After computing the value we get $\mathcal{KL}(\boldsymbol{\pi}||\hat{\boldsymbol{\pi}}) = 0.003661656$. Thus the two distribution are not that far away from each other in the $\mathcal{KL}$-divergence sense.

2 Web community:

Let's consider a set of Web pages, each of which has two keywords and belongs to one of the following three thematic groups:

- Sports related pages (S)
- Culture related pages (C)
- Beauty care related pages (B)

We assume that surfer surfs on the the websites available in the set of web pages. A path of the web pages surfed can be considered as a sample from a HMM with transition matrix $\mathbf{A}$ between the hidden states sports, culture and beauty given by:

$$\mathbf{A} = \begin{pmatrix} 0.7 & 0.2 & 0.1 \\ 0.25 & 0.7 & 0.05 \\ 0.1 & 0.1 & 0.8 \end{pmatrix}$$

and a transition matrix $\mathbf{B}$ defined by table 1. We will further assume that the two keywords of a web page are conditionally independent given the groups they belong to. Therefore, there exists $10 \times 10 = 100$ couples of keywords possible for a given page. The first exercise of this part will be to simulate from the given model

	Sports	Culture	Beauty
Abs	0.2	0	0.1
Cosmetics	0	0.1	0.3
Books	0.1	0.3	0
Age	0.1	0.2	0.2
Force	0.2	0	0.1
Endurance	0.3	0	0
Resilience	0.1	0.1	0.1
Cream	0	0	0.2
History	0	0.2	0
Mathematics	0	0.1	0

Table 1: The emission matrix by thematic group

above. We will generate 30 samples (sequences) each of length 100 from our model.
First let's begin by noticing that given the above table 1, we can computing the corresponding emission matrix for the couples of key words. To begin let Z belong to the hidden states $\{S, C, B\}$ and $X = (X_1, X_2)$ (abusing notations here) the emitted couple. Where X_1, and X_2 are elements of the rows of table 1. Since given the thematic group the two key-word are independent, we can compute the new emission matrix for the couples. Which can mathematically be computed (for the thematic group S for example):

$$\begin{aligned} P(X = (a, b)|Z = S) =& \mathbb{P}(X_1 = a, X_2 = b|Z = S) \\ &= \mathbb{P}(X_1 = a|Z = S)P(X_2 = b|Z = S) \end{aligned}$$

This will be easy to compute for all of couples and the given the other hidden states C and B. In doing so we will get a new emission matrix of size 100×3. Nonetheless, let's notice that this is the case if the couples (a, b) and (b, a) are seen as distinct, but we will view them as the same couple. However, we have to track the right rows in the 100×3 emission matrix and only keep one of the couples $((a, b), (b, a))$ and double the probability of the kept couple. This process explained above will reduce the 100 rows of the emission matrix to $100 - \sum_{i=1}^{i=9} i$. Then we are left with 55 rows since $\sum_{i=1}^{i=9} i = 45$. The final emission matrix for the couples will be of size 55×3, and it is a big matrix, so we will not show it in the report. However, it will be available in the accompanying codes.
Generating from HMM is done in two steps. First we generate the hidden states (thematic groups) according to the transition matrix , and then given these states we generate the samples (couples of keywords) according to the emission matrix. This can be shown in the figure below:

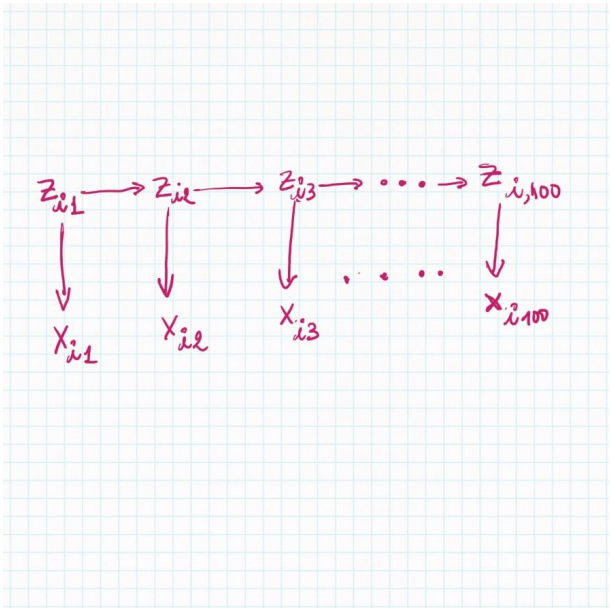

Figure 2: A graphical representation of the HMM

To make matters simple for implementing the generating process in R, we make a naming choices. First we let the thematic groups be $\{1, 2, 3\}$ instead of $\{S, C, B\}$, and we also note that since we have 55 couples of keywords we will name them $\{1, 2, 3, 4, 5, ..., 55\}$ where 1 for the couples corresponds to (Abs, Abs), 2 to (Abs, Cosmetics), 3 to (Abs, Books), and so on so forth. Where, when we get to next couples beginning with (Cosmetics,), we skip (Cosmetics, Abs) and begin with (Cosmetics, Cosmetics), and so on and so forth. We are ready to sample from our model. We fix the initial probabilities for the hidden states to be $\boldsymbol{\pi} = \begin{bmatrix} 0.1 & 0.3 & 0.6 \end{bmatrix}^\top$

Algorithm 1 An algorithm to sample from the HMM

Require: n , T , $\mathbf{A}$, $\mathbf{B}$

$Z \leftarrow \mathbf{Matrix}(0, size = n \cdot T, nrow = n)$
$X \leftarrow \mathbf{Matrix}(0, size = n \cdot T, nrow = n)$
$\boldsymbol{\pi} \leftarrow Vect(\begin{bmatrix} 0.1 & 0.3 & 0.6 \end{bmatrix})$
for i *in* $1 : n$ **do**
  $Z[i, 1] \leftarrow sample\ (1 : 3,\ size{=}1,\ prob{=}\boldsymbol{\pi})$
  $X[i, 1] \leftarrow sample\ (1 : 55,\ size{=}1,\ prob{=}\mathbf{B}[,\ Z[i, 1]])$
end for
for i *in* $1 : n$ **do**
  for t *in* $2 : T$ **do**
    $Z[i, t] \leftarrow sample\ (1 : 3,\ size{=}1,\ prob{=}\mathbf{A}[Z[i,\ t - 1],])$
    $X[i, t] \leftarrow sample\ (1 : 55,\ size{=}1,\ prob{=}\mathbf{B}[,\ Z[i,\ t]])$
  end for
end for

After having sampled from the HMM (30 sequences each of length 100 in our case) one interesting problem is to try and recover the parameters that generated the samples. We first notice that a usual method for these kinds of problems (estimating parameters of a model given samples from the model) is the Maximum Likelihood Estimator (MLE). However, it is easy to notice that the hidden states are never observed . Therefore, we have an incomplete likelihood. We, therefore, result to the Expectation Maximization (EM) algorithm.This algorithm in the case of the HMM is known as the Baum-Welch algorithm. The derivations will be detailed in Appendix

B. Bellow is the algorithm:

Algorithm 2 Baum-Welch algorithm

Require: $\boldsymbol{\pi}$, $\mathbf{A}$, $\mathbf{B}$, **data**, n_{iter}, ϵ

$\mathbf{A}^{old} \leftarrow \mathbf{A}$
$\mathbf{A}^{old} \leftarrow \mathbf{B}$
$\boldsymbol{\pi}^{old} \leftarrow \boldsymbol{\pi}$
for *iter in* $1 : n_{iter}$ **do**
  for k *in* $1 : K$ **do**
    $\pi_k^{new} \leftarrow$ **update** $[\boldsymbol{pi}](\boldsymbol{\pi}^{old}, \mathbf{A}^{old}, \mathbf{B}^{old})$ using B.10.5
  end for
  for ℓ *in* $1 : K$ **do**
    for k *in* $1 : K$ **do**
      $A_{\ell k}^{new} \leftarrow$ **update** $[\mathbf{A}](\boldsymbol{\pi}^{old}, \mathbf{A}^{old}, \mathbf{B}^{old})$ using B.10.6
    end for
  end for
  for m *in* $1 : M$ **do**
    for k *in* $1 : K$ **do**
      $B_{mk}^{new} \leftarrow$ **update** $[\mathbf{B}](\boldsymbol{\pi}^{old}, \mathbf{A}^{old}, \mathbf{B}^{old})$ using B.10.7
    end for
  end for
  $\mathbf{A}^{old} \leftarrow \mathbf{A}^{new}$
  $\mathbf{A}^{old} \leftarrow \mathbf{B}^{new}$
  $\boldsymbol{\pi}^{old} \leftarrow \boldsymbol{\pi}^{new}$
Stop if stoping criterion met
end for

After having implemented the algorithm, we went ahead and estimated the parameters. This procedure can be found in the accompanying codes. Now what we can do is to try and estimate the sequence of the hidden states of the first simulation. The algorithm that does this process is known as the Viterbi algorithm. Since it had already been derived and implemented in class, we use it without derivation. Note we generated a 30 sequences of length 100 each, so the first sequence of hidden state is of length 100. After calculating the error rate using the optimal parameters we get a value of 6. However, on the estimated parameters (using any initialization) it appears to be doing bad. On a good note, the log likelihood is always increasing. We thus initialized the parameters using a noisy perturbation of the optimal parameters, and found with a small to moderate noise level the algorithm is performing rather well. This part is viewable in the accompanying codes.

3 Community Detection

In this section we are presented a set of $n = 90$ web pages each of which belongs to one the three following groups:

- Sports related pages (S)
- Culture related pages (C)
- Beauty care related pages (B)

We will simulate an adjacency matrix of this directed graph where each node corresponds to a web page (belonging to one of the three groups), and the directed edges are the links between the pages. The simulations will be done using a SBM defined below:

- $X_{ij}|z_i = k, z_j = \ell \sim \mathcal{B}(\alpha \mathbb{I}_{(k=\ell)} + \beta \mathbb{I}_{(k\neq\ell)})$
- $\forall k, \mathbb{P}(z_i = k) = \pi_k = \frac{1}{3}$ with $\alpha = 0.15$ and $\beta = 0.05$

Simulating the adjacency matrix will follow in 2 steps. First we generate the classes of the 90 pages then we sample the edges according to the above description. First let:

$$\mathbf{\Gamma} = \begin{pmatrix} \alpha & \beta & \beta \\ \beta & \alpha & \beta \\ \beta & \beta & \beta \end{pmatrix}$$

Then above sampling distribution is equivalent to:

- $X_{ij}|z_i = k, z_j = \ell \sim \mathcal{B}(\Gamma_{k\ell})$
- $\forall k, \mathbb{P}(z_i = k) = \pi_k = \frac{1}{3}$ with $\alpha = 0.15$ and $\beta = 0.05$

or even better

- $X_{ij}|z_i, z_j \sim \mathcal{B}(\Gamma[z_i, z_j])$
- $\forall k, \mathbb{P}(z_i = k) = \pi_k = \frac{1}{3}$ with $\alpha = 0.15$ and $\beta = 0.05$

Thus sampling from the model can be describe in the algorithm bellow

Algorithm 3 An algorithm that simulates the adjacency matrix from the SBM

Require: $n\,, \boldsymbol{\pi}\,, \alpha, \beta$

```
K ← length(π)
Γ ← define[Γ]
Z ← Vect(0, size = n)
X ← Matrix(0, size = n · n, nrow = n)
Z ← sample (1 : K, size=n, prob=π)
for i in 1 : n do
    for j in 1 : n do
        if i ≠ j then
            X[i, j] ← rbinom (n = 1, size=1, prob=Γ[Z[i], Z[j]])
        end if
    end for
end for
```

We used the above described algorithm to sample from the model using the values of α and γ above and below is a figure of the graph corresponding to the adjacency matrix and the web classes

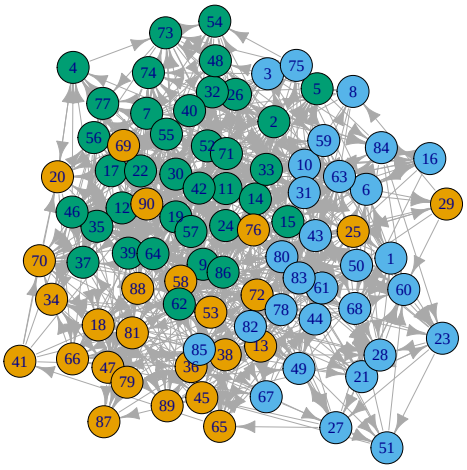

Figure 3: A directed Graph simulated using a Stochastic Block Model

3.1 Simulating a Surf Path from Two Classes of Transition Matrices:

Consider,

$$A^1_{ij} = \frac{X_{ij} + \epsilon}{\sum_j X_{ij} + n\epsilon} \text{ with } \epsilon = \frac{1}{1000}$$

and,

$$A^2_{ij} = \frac{1}{n}$$

We will simulate two sequences each of length 500 describing the path followed by a surfer using A^1, and A^2. A path in this case is a sequences of couples of key words (the emission probabilities of the couple under the groups that the website belongs is given by table1)

Algorithm 4 An algorithm that simulates from the process described above

Require: $\mathbf{A}, \mathbf{B}$

group ← **clusters** *the classes that generated the node classes*
webs ← $Vect(0, size = n)$
couples ← $Vect(0, size = n)$
webs[1] ← 7 *Beging at website 7*
couples[1] ← *sample* $(1 : 55,$ *size*=1, *prob*=**B**$[, \mathbf{group}[\mathbf{webs}[1]]])$
for i *in* $2 : 500$ **do**
  webs$[i]$ ← *sample* $(1 : 90,$ *size*=1, *prob*=**A**$[\mathbf{webs}[i-1],])$
  couples$[i]$ ← *sample* $(1 : 55,$ *size*=1, *prob*=**B**$[, \mathbf{group}[\mathbf{webs}[i]]])$
end for

Appendix

A Deriving the MLE estimator for the Transition Matrix In the Markov Model

Let $\boldsymbol{\theta}$ be the parameters $\boldsymbol{\theta} = (\boldsymbol{\pi}, \mathbf{A})$. Let $x_{1:T}$ be our sequence. The likelihood of the sequence given $\boldsymbol{\theta}$ is

$$\begin{aligned} p(x_{1:T}|\boldsymbol{\theta}) &= p(x_1|\boldsymbol{\theta})p(x_2|x_1,\theta)\dots p(x_T|x_{1:T-1},\boldsymbol{\theta}) \\ &= p(x_1|\boldsymbol{\theta})\prod_{t=2}^{T} p(x_t|x_{1:t-1},\theta) \end{aligned}$$

Given the Markov property, we know that(We will in the subsequent line go from $t = 1, t = T-1$, and shift the indices) : $p(x_t|x_{1:t-1},\theta) = p(x_t|x_{t-1},\boldsymbol{\theta})$ It, thus, follows that

$$\begin{aligned} p(x_{1:T}|\boldsymbol{\theta}) &= p(x_1|\boldsymbol{\theta})\prod_{t=2}^{T} p(x_t|x_{t-1},\boldsymbol{\theta}) \\ &= \prod_{j=1}^{K} p(x_1=j)^{\mathbb{I}(x_1=j)} \prod_{t=1}^{T-1}\prod_{j=1}^{K}\prod_{k=1}^{K} p(x_{t+1}=k|x_t=j)^{\mathbb{I}(x_{t+1}=k,x_t=j)} \\ &= \prod_{j=1}^{K} p(x_1=j)^{\mathbb{I}(x_1=j)} \prod_{t=1}^{T-1}\prod_{j=1}^{K}\prod_{k=1}^{K} A_{jk}^{\mathbb{I}(x_{t+1}=k,x_t=j)} \\ &= \prod_{j=1}^{K} (\pi_j)^{\mathbb{I}(x_1=j)} \prod_{t=1}^{T-1}\prod_{j=1}^{K}\prod_{k=1}^{K} A_{jk}^{\mathbb{I}(x_{t+1}=k,x_t=j)} \end{aligned}$$

We take the log and get the log likelihood below

$$p(x_{1:T}|\boldsymbol{\theta}) = \sum_j \mathbb{I}(x_1=j)\log\pi_j + \sum_j\sum_k N_{jk}\log A_{jk}$$

where we define the following count:

$$N_{jk} = \sum_{i=1}^{N}\sum_{t=1}^{T-1} \mathbb{I}(x_t=j, x_{t+1}=k)$$

For the estimation of the transition matrix estimation c does not depend on the transition matrix:

$$\log p(D \mid \theta) = c + \sum_j^K\sum_k^K N_{jk}\log A_{jk}$$

where $c = \sum_{j=1}^{K}\mathbb{I}(x_1=j)\log\pi_j$.
To maximize the the log likelihood w.r.t to $\mathbf{A}$ we solve a constrained optimization problem. We have the constraint that $\sum_k^K A_{jk} = 1\,, \forall j \in \{1,2,\cdots K\}$. Therefore K equality constraints. We write the lagrangian with multiplier $\boldsymbol{\lambda}$ a $K-$dimensional vector

$$\mathcal{L}(\mathbf{A},\boldsymbol{\lambda}) = c + \sum_j^K\sum_k^K N_{jk}\log A_{jk} + \sum_{j=1}^{K}\Big[\lambda_j\big(\sum_{k=1}^{K} A_{jk} - 1\big)\Big] \tag{A.0.1}$$

Taking the partials with respect to A_{jk} and $_k$ and setting them to zero we get (We skip this routine here it, for it is very similar those done in B.3, and just give the answer):

$$\hat{A}_{jk} = \frac{N_{jk}}{\sum_k N_{jk}} \tag{A.0.2}$$

B Baum-Welch Derivation

We Begin by deriving the algorithm:
To make matters simple let's first look at one single sequence of length T. let X be the observable, Z be the hidden states and $\boldsymbol{\theta} = (\boldsymbol{\pi},\ \mathbf{A},\ \mathbf{B})$ be the parameters. Then the complete likelihood can be written as (we omit $\boldsymbol{\theta}$ while writing the formulas):

$$\begin{aligned}
p(X_{1:T}, Z_{1:T}|\boldsymbol{\theta}) &= p(Z_1)p(Z_2|Z_1)p(Z_3|Z_1, Z_2)\cdots p(Z_T|Z_{1:T-1}) \\
&\quad p(X_1|Z_{1:T})p(X_2|Z_{1:T}, X_1)p(X_2|Z_{1:T}, X_{1:2})\cdot p(X_T|Z_{1:T}, X_{1:T-1}) \\
&= p(Z_1)\prod_{t=1}^{T-1} p(Z_{t+1}|Z_t)\prod_{t=1}^{T} p(X_t|Z_t) \textit{ follows form the Markov property} \\
&= \prod_{k=1}^{K} p(Z_1 = k)^{\mathbb{I}(Z_1=k)} \prod_{t=1}^{T-1}\prod_{k=1}^{K}\prod_{\ell=1}^{\ell=K} p(Z_{t+1} = k|Z_t = \ell)^{\mathbb{I}(Z_{t+1}=k, Z_t=\ell)} \prod_{t=1}^{t=T}\prod_{k=1}^{K} p(X_t|Z_t = k)^{\mathbb{I}(Z_t=k)}
\end{aligned} \tag{B.0.1}$$

Now take the log of the complete likelihood assuming observed samples $(x_{1:n},\ z_{1:n})$ Eq(B.0.1) we get

$$\begin{aligned}
\log p(x_{1:T}, z_{1:T}|\boldsymbol{\theta}) &= \sum_{k=1}^{K} \mathbb{I}(z_1 = k)\log p(z_1 = k) \\
&\quad + \sum_{t=1}^{T-1}\sum_{k=1}^{K}\sum_{\ell=1}^{\ell=K} \mathbb{I}(z_{t+1} = k, z_t = \ell)\log p(z_{t+1} = k|z_t = \ell) \\
&\quad + \sum_{t=1}^{t=T}\sum_{k=1}^{K} \mathbb{I}(z_t = k)\log p(x_t|z_t = k) \\
&= \sum_{k=1}^{K} \mathbb{I}(z_1 = k)\log \pi_k + \sum_{t=1}^{T-1}\sum_{k=1}^{K}\sum_{\ell=1}^{\ell=K} \mathbb{I}(z_{t+1} = k, z_t = \ell)\log A_{\ell k} \\
&\quad + \sum_{t=1}^{t=T}\sum_{k=1}^{K} \mathbb{I}(z_t = k)\log B_{x_t}(k)
\end{aligned} \tag{B.0.2}$$

B.1 Deriving the objective

We are ready to compute the E and M steps of the EM-algorithm. In the E-step we compute $\mathbb{E}[\log p(x_{1:T}, z_{1:T}|\boldsymbol{\theta}^s)]$, call it $\mathcal{Q}(\boldsymbol{\theta},\ \boldsymbol{\theta}^s)$ under the distribution of $Z_{1:T}|X_{1:T}$. Notice that taking the expectation in Eq(B.0.2) will pass through the sum by linearity and only the indicator variables will remain. Thus their expected values will become probabilities:

$$\begin{aligned}
\mathcal{Q}(\boldsymbol{\theta},\ \boldsymbol{\theta}^s) &= \sum_{k=1}^{K} p(z_1 = k|x_{1:T},\ \boldsymbol{\theta}^s)\log \pi_k + \sum_{t=1}^{T-1}\sum_{k=1}^{K}\sum_{\ell=1}^{\ell=K} p(z_{t+1} = k, z_t = \ell|x_{1:T},\ \boldsymbol{\theta}^s)\log A_{\ell k} \\
&\quad + \sum_{t=1}^{T}\sum_{k=1}^{K} p(z_t = k|x_{1:T},\ \boldsymbol{\theta}^s)\log B_{x_t}(k)
\end{aligned} \tag{B.1.1}$$

Now one simple remark we do is that this was for one single sequence since we have 30 sequences, we could first note that total log-likelihood for all the sequences (independent samples on the number of sequences). Meaning instead of observing $(x_{1:T}, x_{1:T})$ we have $(x_{i1:iT}, x_{i1:iT})$. Thus for our problem, $\mathcal{Q}(\boldsymbol{\theta},\, \boldsymbol{\theta}^s)$ is :

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\theta},\, \boldsymbol{\theta}^s) &= \sum_{i=1}^{n}\sum_{k=1}^{K} p(z_{i1}=k|x_{i1:iT},\, \boldsymbol{\theta}^s)\log \pi_k \\ &+ \sum_{i=1}^{n}\sum_{t=1}^{T-1}\sum_{k=1}^{K}\sum_{\ell=1}^{\ell=K} p(z_{i(t+1)}=k, z_{it}=\ell|x_{i1:iT},\, \boldsymbol{\theta}^s)\log A_{\ell k} \\ &+ \sum_{i=1}^{n}\sum_{t=1}^{T}\sum_{k=1}^{K} p(z_{it}=k|x_{i1:iT},\, \boldsymbol{\theta}^s)\log B_{x_{it}}(k) \end{aligned} \tag{B.1.2}$$

Note that in the equation above one might ask why doesn't the terms with the parameters have i's inside them. They should technically have i's in them. However we can notice that $p(z_{i1}=k)=\pi_k$ irrespective of which of the sequence we ware looking at, and the same can be said of $p(z_{i(t+1)}=k|z_{it}=\ell)=A_{lk}$. Now for the M-step the we have to find the $\boldsymbol{\theta}$ call it $\boldsymbol{\theta}^{(s+1)}$ that maximizes $\mathcal{Q}(\boldsymbol{\theta},\, \boldsymbol{\theta}^s)$ from Eq(B.1.2). To solve for the maximum we can first begin by writing the Lagrangian of the problem. Note that this require the constraints $\sum_k \pi_k = 1$, $\sum_{k=1}^{K} A_{\ell k} = 1\,, \forall \ell \in \{1,2,\cdots,K\}$, and $\sum_{m=1}^{M} B_m(k) = 1\,, \forall k \in \{1,2,\cdots,K\}$. We will thus use the following Lagrange multipliers. $\lambda \in \mathbb{R}$, $\boldsymbol{\eta}$ a $K-$dimensional vector, and $\boldsymbol{\beta}$ a $K-$dimensional vector.

$$\mathcal{L}(\boldsymbol{\theta},\, \boldsymbol{\theta}^s, \lambda, \boldsymbol{\eta}, \boldsymbol{\beta}) = \mathcal{Q}(\boldsymbol{\theta},\, \boldsymbol{\theta}^s) + \lambda(\sum_{k=1}^{K}\pi_k - 1) + \sum_{\ell=1}^{K}\Big[\eta_\ell(\sum_{k=1}^{K} A_{\ell k} - 1)\Big] + \sum_{k=1}^{K}\Big[\beta_k(\sum_{m=1}^{M} B_m(k) - 1)\Big] \tag{B.1.3}$$

Where m in the above Eq(B.1.3) is the possible values emitted in our case $m \in \{1,2,...,55\}$ Now to solve for the arg max, we take the partial derivatives and set them to zero.
First, with respect to π.

B.2 The Updates for The Inital Probabilities

To compute the partial derivatives, we take the indices to be ℓ_o and k_o. This just to avoid confusion between what is changing and what is fixed. With $\ell_o \in \{1,2,...K\}$, and $k_o \in \{1,2,...K\}$:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \pi_{k_o}} &= \frac{\partial}{\partial \pi_{k_o}}\Big(\sum_{i=1}^{n}\sum_{k=1}^{K} p(z_{i1}=k|x_{i1:iT},\, \boldsymbol{\theta}^s)\log \pi_k + \lambda(\sum_{k=1}^{K}\pi_k - 1)\Big) \\ &= \frac{\sum_{i=1}^{n} p(z_{i1}=k_o|x_{i1:iT},\, \boldsymbol{\theta}^s)}{\pi_{k_o}} + \lambda \end{aligned}$$

Thus solving for π_{k_o} by setting $\frac{\partial \mathcal{L}}{\partial \pi_{k_o}}$ to 0 we get:

$$\hat{\pi}_{k_o} = -\frac{\sum_{i=1}^{n} p(z_{i1}=k_o|x_{i1:iT},\, \boldsymbol{\theta}^s)}{\lambda} \tag{B.2.1}$$

We also take the partials with respect to λ and we get:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \lambda} &= \frac{\partial}{\partial \lambda}\Big(\lambda(\sum_{k=1}^{K}\pi_k - 1)\Big) \\ &= \sum_{k=1}^{K}\pi_k - 1 \end{aligned} \tag{B.2.2}$$

And setting it to zero we get:

$$\sum_{k=1}^{K} \hat{\pi}_{k_o} = 1$$

It then follows that:

$$\sum_{k_o=1}^{K} \pi_k = 1$$

Therefore combining the above equation with B.2.1 we get:

$$\sum_{k_o=1}^{K} \hat{\pi}_{k_o} = -\frac{\sum_{k_o=1}^{K}\sum_{i=1}^{n} p(z_{i1} = k_o | x_{i1:iT}, \boldsymbol{\theta}^s)}{\lambda} = 1$$

Thus :

$$\lambda = -\sum_{k_o=1}^{K}\sum_{i=1}^{n} p(z_{i1} = k_o | x_{i1:iT}, \boldsymbol{\theta}^s) = -n \tag{B.2.3}$$

To see why the above expression is true, note the summands are probabilities. Over k_o they will sum to 1 and over i it then becomes n. Replacing the value of λ from Eq(B.2.3) in to Eq(B.2.1) we finally get the updates for $\boldsymbol{\pi}$:

$$\hat{\pi}_{k_o}^{(s+1)} = \frac{\sum_{i=1}^{n} p(z_{i1} = k_o | x_{i1:iT}, \boldsymbol{\theta}^s)}{n} \quad \forall k_o \in \{1, 2, \cdots, K\} \tag{B.2.4}$$

B.3 The Updates for The Transition Probabilities

Now for the update equations for $A_{\ell k}$, we also do the same procedure take the partial derivatives of $\mathcal{L}$ and set them 0. To compute the partial derivatives, we take the indices to be ℓ_o and k_o. This just to avoid confusion between what is changing and what is fixed. With $\ell_o \in \{1, 2, ...K\}$, and $k_o \in \{1, 2, ...K\}$:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial A_{\ell_o k_o}} &= \frac{\partial}{\partial A_{\ell_o k_o}} \left(\sum_{i=1}^{n}\sum_{t=1}^{T-1}\sum_{k=1}^{K}\sum_{\ell=1}^{\ell=K} p(z_{i(t+1)} = k, z_{it} = \ell | x_{i1:iT}, \boldsymbol{\theta}^s) \log A_{\ell k} + \sum_{\ell=1}^{K} \Big[\eta_\ell \big(\sum_{k=1}^{K} A_{\ell k} - 1 \big) \Big] \right) \\
&= \frac{\sum_{i=1}^{n}\sum_{t=1}^{T-1} p(z_{i(t+1)} = k_o, z_{it} = \ell_o | x_{i1:iT}, \boldsymbol{\theta}^s)}{A_{\ell_o k_o}} + \eta_{\ell_o}
\end{aligned}$$

and by setting the above equation to 0 we get:

$$\hat{A}_{\ell_o k_o} = -\frac{\sum_{i=1}^{n}\sum_{t=1}^{T-1} p(z_{i(t+1)} = k_o, z_{it} = \ell_o | x_{i1:iT}, \boldsymbol{\theta}^s)}{\eta_{\ell_o}} \tag{B.3.1}$$

We also take the partials with respect to η and we get:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \eta_{\ell_o}} &= \frac{\partial}{\partial \eta_{\ell_o}} \Big(\sum_{\ell=1}^{K} \Big[\eta_\ell \big(\sum_{k=1}^{K} A_{\ell k} - 1 \big) \Big] \Big) \Big) \\
&= \sum_{k=1}^{K} A_{\ell_o k} - 1
\end{aligned}$$

Thus after setting the above equation to zero we get:

$$\sum_{k=1}^{K} A_{\ell_o k} = 1$$

It follows that:

$$\sum_{k_o=1}^{K} \hat{A}_{\ell_o k_o} = 1$$

Combining the above equation with B.3.1 we get:

$$\sum_{k_o=1}^{K} \hat{A}_{\ell_o k_o} = -\frac{\sum_{k_o=1}^{K}\sum_{i=1}^{n}\sum_{t=1}^{T-1} p(z_{i(t+1)} = k_o, z_{it} = \ell_o | x_{i1:iT}, \boldsymbol{\theta}^s)}{\eta_{\ell_o}} = 1$$

Solving the above equation for η_{ell_o} we get:

$$\eta_{\ell_o} = -\sum_{k_o=1}^{K}\sum_{i=1}^{n}\sum_{t=1}^{T-1} p(z_{i(t+1)} = k_o, z_{it} = \ell_o | x_{i1:iT}, \boldsymbol{\theta}^s)$$

We will simplify the above equation a little more. First we take the first sum with respect to k_o, and note that this is equal to a marginalization. We will end up with $p(z_{it} = \ell_o | x_{i1:iT}, \boldsymbol{\theta}^s)$. Thus we get

$$\eta_{\ell_o} = -\sum_{i=1}^{n}\sum_{t=1}^{T-1} p(z_{it} = \ell_o | x_{i1:iT}, \boldsymbol{\theta}^s)$$

Combining the above equation with Eq(B.3.1) we get the updating equation as:

$$\hat{A}^{(s+1)}_{\ell_o k_o} = \frac{\sum_{i=1}^{n}\sum_{t=1}^{T-1} p(z_{i(t+1)} = k_o, z_{it} = \ell_o | x_{i1:iT}, \boldsymbol{\theta}^s)}{\sum_{i=1}^{n}\sum_{t=1}^{T-1} p(z_{it} = \ell_o | x_{i1:iT}, \boldsymbol{\theta}^s)} \quad \forall \ell_o, k_o \in \{1, 2, \cdots, K\} \tag{B.3.2}$$

B.4 The Updates for The Emission Probabilities

We now get to the last update parameters. To conform to the notation in the report, our notation follows that the columns of B are the states on to which we are conditioning and the rows are the couples to be emitted (i.e $B[i, j] = p(x = i | z = j)$). As in the previous parts we take the partial derivatives and set them to zero:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial B_{m_o}(k_o)} &= \frac{\partial}{\partial B_{m_o}(k_o)} \left(\sum_{i=1}^{n}\sum_{t=1}^{T}\sum_{k=1}^{K} p(z_{it} = k | x_{i1:iT}, \boldsymbol{\theta}^s) \log B_{x_{it}}(k) + \sum_{k=1}^{K} \left[\beta_k \Big(\sum_{m=1}^{M} B_m(k) - 1 \Big) \right] \right) \\
&= \frac{\partial}{\partial B_{m_o}(k_o)} \left(\sum_{i=1}^{n}\sum_{t=1}^{T}\sum_{k=1}^{K} p(z_{it} = k | x_{i1:iT}, \boldsymbol{\theta}^s) \log B_{x_{it}}(k) \right) + \beta_{k_o}
\end{aligned}$$

Now to be able to compute the above partial derivative with respect to $B_{m_o}(k_o)$. We will add a 1 in the form of the sum of an indicator variable we get:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial B_{m_o}(k_o)} &= \frac{\partial}{\partial B_{m_o}(k_o)} \left(\sum_{i=1}^{n}\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{m=1}^{M} p(z_{it} = k | x_{i1:iT}, \boldsymbol{\theta}^s) \log B_m(k) \mathbb{I}(x_{it} = m) \right) + \beta_{k_o} \\
&= \frac{\sum_{i=1}^{n}\sum_{t=1}^{T} p(z_{it} = k_o | x_{i1:iT}, \boldsymbol{\theta}^s) \mathbb{I}(x_{it} = m_o)}{B_{m_o}(k_o)} + \beta_{k_o}
\end{aligned}$$

and setting the above equation to zero and solving for $B_{m_o}(k_o)$ we get:

$$\hat{B}_{m_o}(k_o) = -\frac{\sum_{i=1}^{n}\sum_{t=1}^{T} p(z_{it} = k_o | x_{i1:iT}, \boldsymbol{\theta}^s)\mathbb{I}(x_{it} = m_o)}{\beta_{k_o}} \tag{B.4.1}$$

Now we compute the partial with respect to β_{k_o}. Ignoring terms that are constant w.r.t β_{k_o} we get:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \beta_{k_o}} &= \frac{\partial}{\partial \beta_{k_o}}\left(\sum_{k=1}^{K}\left[\beta_k\Big(\sum_{m=1}^{M} B_m(k) - 1\Big)\right]\right) \\ &= \sum_{m=1}^{M} B_m(k_o) - 1\end{aligned}$$

and setting the above equation to zero we get:

$$\sum_{m=1}^{M} B_m(k_o) = 1$$

It thus follows that:

$$\sum_{m_o=1}^{M} \hat{B}_{m_o}(k_o) = 1$$

Now combining the above equation with Eq(B.4.1) we get:

$$\begin{aligned}\sum_{m_o=1}^{M} \hat{B}_{m_o}(k_o) &= -\frac{\sum_{m_o=1}^{M}\sum_{i=1}^{n}\sum_{t=1}^{T} p(z_{it} = k_o | x_{i1:iT}, \boldsymbol{\theta}^s)\mathbb{I}(x_{it} = m_o)}{\beta_{k_o}} \\ &= -\frac{\sum_{i=1}^{n}\sum_{t=1}^{T} p(z_{it} = k_o | x_{i1:iT}, \boldsymbol{\theta}^s)\sum_{m_o=1}^{M}\mathbb{I}(x_{it} = m_o)}{\beta_{k_o}} \\ &= -\frac{\sum_{i=1}^{n}\sum_{t=1}^{T} p(z_{it} = k_o | x_{i1:iT}, \boldsymbol{\theta}^s)}{\beta_{k_o}}\end{aligned}$$

The last equality is from the observation that for a give i and t $\sum_{m_o=1}^{M}\mathbb{I}(x_{it} = m_o) = 1$, for an observation is indeed unique (it cannot two different m_o's at same time). Thus

$$\sum_{m_o=1}^{M} \hat{B}_{m_o}(k_o) = 1 \implies -\frac{\sum_{i=1}^{n}\sum_{t=1}^{T} p(z_{it} = k_o | x_{i1:iT}, \boldsymbol{\theta}^s)}{\beta_{k_o}} = 1$$

and we finally have:

$$\beta_{k_o} = -\sum_{i=1}^{n}\sum_{t=1}^{T} p(z_{it} = k_o | x_{i1:iT}, \boldsymbol{\theta}^s) \tag{B.4.2}$$

Plugging B.4.2 into B.4.1 we obtain the updates for the emmission matrix as:

$$\hat{B}_{m_o}^{(s+1)}(k_o) = \frac{\sum_{i=1}^{n}\sum_{t=1}^{T} p(z_{it} = k_o | x_{i1:iT}, \boldsymbol{\theta}^s)\mathbb{I}(x_{it} = m_o)}{\sum_{i=1}^{n}\sum_{t=1}^{T} p(z_{it} = k_o | x_{i1:iT}, \boldsymbol{\theta}^s)} \tag{B.4.3}$$

B.5 Derivation of the helpers needed to implement the Baum-Welch:

B.6 The Forward algorithm

We do this for a single sequence it will be straight forward for the multiple sequences. Let $\alpha_t(k) = p(z_t = k, x_{1:t})$, then we can compute a recursive expression for $\alpha_t(k)$ in the following:

$$\begin{aligned}
\alpha_t(k) &= p(z_t = k, x_{1:t}) \qquad \text{(B.6.1)} \\
&= \sum_{\ell=1}^{K} p(x_{1:t-1}, x_t, z_t = k, z_t = \ell) \\
&= \sum_{\ell=1}^{K} p(x_t, z_t = k | x_{1:t-1}, z_{t_1} = \ell) p(x_{1:t-1}, z_t = \ell) \\
&= \sum_{\ell=1}^{K} p(z_t = k | x_{1:t-1}, z_{t_1} = \ell) p(x_t | z_t = k, x_{1:t-1}, z_t = \ell) p(x_{1:t-1}, z_t = \ell) \\
&= \sum_{\ell=1}^{K} p(z_t = k | z_t = \ell) p(x_t | z_t = k) p(x_{1:t-1}, z_{t_1} = \ell) \text{ ,by the conditional indep in the HMM} \\
&= \sum_{\ell=1}^{K} A_{\ell k} B_{x_t}(k) \alpha_{t-1}(\ell) \qquad \text{(B.6.2)}
\end{aligned}$$

The above derivation is know as the forward algorithm. It is an iterative algorithm. thus $\alpha_1(k)$ is required, and by definition:

$$\begin{aligned}
\alpha_1(k) &= p(z_1 = k, x_{1:1}) \\
&= p(z_1 = k, x_{1:1}) \\
&= p(z_1 = k) p(x_1 | z_1 = k) \\
&= \pi_k B_{x_1}(k)
\end{aligned}$$

B.7 The Backward Algorithm

Let $\beta_t(k) = p(x_{(t+1):T} | z_t = k)$, then we can compute a recursive expression for $\alpha_t(k)$ in the following manner:

$$\begin{aligned}
\beta_t(k) = p(x_{(t+1):T} | z_t = k) &= \sum_{\ell=1}^{K} p(x_{t+1}, x_{(t+2):T}, z_{t+1} \\
&= \ell | z_t = k) \\
&= \sum_{\ell=1}^{K} p(z_{t+1} = \ell | z_t = k) p(x_{t+1}, x_{(t+2):T} | z_{t+1} = \ell, z_t = k) \\
&= \sum_{\ell=1}^{K} p(z_{t+1} = \ell | z_t = k) p(x_{t+1} | z_{t+1} = \ell, z_t = k) p(x_{(t+2):T} | z_{t+1} = \ell, z_t = k, x_{t+1}) \\
&= \sum_{\ell=1}^{K} p(z_{t+1} = \ell | z_t = k) p(x_{t+1} | z_{t+1} = \ell) p(x_{(t+2):T} | z_{t+1} = \ell) \text{ ,the HMM Factorization} \\
&= \sum_{\ell=1}^{K} A_{k\ell} B_{x_{t+1}}(\ell) \beta_{t+1}(k) \qquad \text{(B.7.1)}
\end{aligned}$$

The above derivation is know as the backward algorithm. It is an iterative algorithm, and $\beta_T(k) = 1 \, , \forall k$

B.8 Tau and Xi

Now that we have the forward and the backward most of the computation neede for the Baum-Welch will follow nicely(we are still focusing on a single sequence, generalizing it will come naturally): Let's look at $\tau_t(k) = p(z_t = k|x_{1:T}, \boldsymbol{\theta}^s)$. We will omit $\boldsymbol{\theta}^s$ (keeping in mind that it is in there).

$$\begin{aligned}
\tau_t(k) &= \frac{p(z_t = k, x_{1:T})}{p(x_{1:T})} \\
&= \frac{p(z_t = k, x_{1:t}, x_{(t+1):T})}{p(x_{1:T})} \\
&= \frac{p(z_t = k, x_{1:t})p(x_{(t+1):T}|z_t = k, x_{1:t})}{p(x_{1:T})} \\
&= \frac{p(z_t = k, x_{1:t})p(x_{(t+1):T}|z_t = k)}{p(x_{1:T})} \text{ by the conditional indep of HMM} \\
&= \frac{\alpha_t(k)\beta_t(k)}{p(x_{1:T})} \qquad \text{(B.8.1)}
\end{aligned}$$

and we can also note that:

$$p(x_{1:T}) = \sum_{k=1}^{K} p(x_{1:T}, z_T = k) = \sum_{k=1}^{K} \alpha_T(k) \qquad \text{(B.8.2)}$$

Finally,

$$\tau_t(k) = \frac{\alpha_t(k)\beta_t(k)}{\sum_{k=1}^{K} \alpha_T(k)} \qquad \text{(B.8.3)}$$

Now we look at the second expression needed for the Baum-Welch (still on a single sequence, and omitting $\boldsymbol{\theta}^s$).
$\xi_{t,t+1}(\ell, k) = p(z_{t+1} = k, z_t = \ell|x_{1:T}; \boldsymbol{\theta}^s)$.
$\forall k \in \{1, 2, \cdots, K\}, \forall \ell \in \{1, 2, \cdots, K\}, \forall t \in \{1, 2, \cdots, T-1\}$

$$\begin{aligned}
\xi_{t,t+1}(\ell, k) &= \frac{p(z_{t+1} = k, z_t = \ell, x_{1:T})}{p(x_{1:T})} \qquad \text{(B.8.4)} \\
&= \frac{p(z_{t+1} = k, z_t = \ell, x_{1:t}, x_{t+1}, x_{(t+2):T})}{p(x_{1:T})} \\
&= \frac{p(x_{1:t}, z_t = \ell)p(z_{t+1} = k, x_{t+1}, x_{(t+2):T}|x_{1:t}, z_t = \ell)}{p(x_{1:T})} \\
&= \frac{p(x_{1:t}, z_t = \ell)p(z_{t+1} = k|x_{1:t}, z_t = \ell)p(x_{t+1}, x_{(t+2):T}|z_{t+1} = k, x_{1:t}, z_t = \ell)}{p(x_{1:T})} \\
&= \frac{p(x_{1:t}, z_t = \ell)p(z_{t+1} = k|x_{1:t}, z_t = \ell)p(x_{t+1}|z_{t+1} = k, x_{1:t}, z_t = \ell)p(x_{(t+2):T}|z_{t+1} = k, x_{1:t}, z_t = \ell, x_{t+1})}{p(x_{1:T})} \\
&= \frac{p(x_{1:t}, z_t = \ell)p(z_{t+1} = k|z_t = \ell)p(x_{t+1}|z_{t+1} = k)p(x_{(t+2):T}|z_{t+1} = k)}{p(x_{1:T})} \text{ by HMM Factorization} \\
&= \frac{\alpha_t(\ell)A_{\ell k}B_{x_{t+1}}(k)\beta_{t+1}(k)}{p(x_{1:T})} \\
&= \frac{\alpha_t(\ell)A_{\ell k}B_{x_{t+1}}(k)\beta_{t+1}(k)}{\sum_{k=1}^{K} \alpha_T(k)} \text{ using Eq(B.8.2) for the denominator} \qquad \text{(B.8.5)}
\end{aligned}$$

Now that we have everything for a single sequence we can generalize it to multiple sequences. This is equivalent to adding an extra dimension (index) that keeps track of which of the sequences we are looking at. Thus we

have for τ, redoing all the steps in B.8.1:

$$\begin{aligned}\tau_{it}(k) &= \frac{p(z_{it}=k, x_{i1:iT})}{p(x_{i1:iT})} \\ &= \frac{\alpha_{it}(k)\beta_{it}(k)}{p(x_{i1:iT})} \\ &= \frac{\alpha_{it}(k)\beta_{it}(k)}{\sum_{k=1}^{K}\alpha_{iT}(k)}\end{aligned} \tag{B.8.6}$$

Note in the single sequence case $\alpha_t(k)$ is a two dimensional array (one dimension for t, and one for k). However, in the multiple sequence case $\alpha_{it}(k)$ is a tensor with 3 dimensions (one for t, one for k, and one i). t tracks the sequence length, k the number of hidden states, and i the number of sequences.

The same follows for ξ. We follow the steps of B.8.4 while track the sequence number:

$$\begin{aligned}\xi_{it,i(t+1)}(\ell,k) &= \frac{p(z_{i(t+1)}=k, z_{it}=\ell, x_{i1:iT})}{p(x_{i1:iT})} \\ &= \frac{\alpha_{it}(\ell)A_{\ell k}B_{x_{i(t+1)}}(k)\beta_{i(t+1)}(k)}{\sum_{k=1}^{K}\alpha_{iT}(k)}\end{aligned} \tag{B.8.7}$$

Here $\xi_{it,i(t+1)}(\ell,k)$ is a four dimensional tensor (i, t, ℓ, k) are the indices.

B.9 Log Sum Exp To Avoid Underflow

Technically we are ready to implement the Baum-Welch algorithm. However, since we will be implementing products of multiple small numbers(probabilities), there is risk of underflow. We will, therefore do the computation $\alpha_{it}(k)$, $\beta_{it}(k)$, and $\xi_{it,i(t+1)}(\ell,k)$ in log probability space (using the Log-Sum-Exponential trick)and go back to probability space just before updating the parameters. This straight forward:
For $\tau_{it}(k)$, taking the log of Eq(B.8.6) we get:

$$\begin{aligned}\log\tau_{it}(k) &= \log\left(\frac{\alpha_{it}(k)\beta_{it}(k)}{\sum_{k=1}^{K}\alpha_{iT}(k)}\right) \\ &= \log\alpha_{it}(k) + \log\beta_{it}(k) - \log\left(\sum_{k=1}^{K}\alpha_{iT}(k)\right) \\ &= \log\alpha_{it}(k) + \log\beta_{it}(k) - \log\left(\sum_{k=1}^{K}\exp\left\{\log\alpha_{iT}(k)\right\}\right)\end{aligned} \tag{B.9.1}$$

For $\xi_{it,i(t+1)}(\ell,k)$, taking the log of Eq(B.8.7) we get:

$$\begin{aligned}\log\xi_{it,i(t+1)}(\ell,k) &= \log\left(\frac{\alpha_{it}(\ell)A_{\ell k}B_{x_{i(t+1)}}(k)\beta_{i(t+1)}(k)}{\sum_{k=1}^{K}\alpha_{iT}(k)}\right) \\ &= \log\alpha_{it}(\ell) + \log A_{\ell k} + \log B_{x_{i(t+1)}}(k) + \log\beta_{i(t+1)}(k) - \log\left(\sum_{k=1}^{K}\alpha_{iT}(k)\right) \\ &= \log\alpha_{it}(\ell) + \log A_{\ell k} + \log B_{x_{i(t+1)}}(k) + \log\beta_{i(t+1)}(k) - \log\left(\sum_{k=1}^{K}\exp\left\{\log\alpha_{iT}(k)\right\}\right)\end{aligned} \tag{B.9.2}$$

Now for $\log \alpha$, we have by taking the log of Eq(B.6.2) and generalizing it to multiple sequences:

$$\begin{aligned}\log \alpha_{it}(k) &= \log\left(\sum_{\ell=1}^{K} A_{\ell k} B_{x_{it}}(k)\alpha_{i(t-1)}(\ell)\right) \\ &= \log\left(\sum_{\ell=1}^{K} \exp\left\{\log A_{\ell k} + \log B_{x_{it}}(k) + \log \alpha_{i(t-1)}(\ell)\right\}\right) \end{aligned} \tag{B.9.3}$$

and same for $log\beta$. We take the log of Eq(B.7.1), and generalizing it to multiple sequences we have:

$$\begin{aligned}\log \beta_{it}(k) &= \log\left(\sum_{\ell=1}^{K} A_{k\ell} B_{x_{t+1}}(\ell)\beta_{t+1}(k))\right) \\ &= \log\left(\sum_{\ell=1}^{K} \exp\left\{\log A_{k\ell} + \log B_{x_{t+1}}(\ell) + \log \beta_{t+1}(k))\right\}\right) \end{aligned} \tag{B.9.4}$$

B.10 A side Note :

We can avoid computing $\tau_{it}(k) = p(z_{it} = k|x_{i1:iT})$ by observing that $\forall t \in \{1, 2, \cdots, T-1\}$:

$$\begin{aligned}\tau_{it}(\ell) &= \sum_{k=1}^{K} \xi_{it,i(t+1)}(\ell, k) \\ &= \sum_{k=1}^{K} p(z_{i(t+1)} = k, z_{it} = \ell|x_{i1:iT}; \boldsymbol{\theta}^s) \\ &= p(z_{it} = \ell|x_{i1:iT}; \boldsymbol{\theta}^s) \;\; \forall \ell \in \{1, 2, \cdots, K\} \end{aligned} \tag{B.10.1}$$

and for $t = T$

$$\begin{aligned}\tau_{iT}(k) &= \sum_{\ell=1}^{K} \xi_{i(T-1),iT}(\ell, k) \\ &= \sum_{\ell=1}^{K} p(z_{iT} = k, z_{i(T-1)} = \ell|x_{i1:iT}; \boldsymbol{\theta}^s) \\ &= p(z_{iT} = k|x_{i1:iT}; \boldsymbol{\theta}^s) \; \forall k \in \{1, 2, \cdots .K\} \end{aligned} \tag{B.10.2}$$

This can be done in log space as well using the log sum exponential trick in the following format:
$\forall t \in \{1, 2, \cdots, T-1\}$ and $\forall \ell \in \{1, 2, \cdots, K\}$

$$\begin{aligned}\log \tau_{it}(\ell) &= \log\left(\sum_{k=1}^{K} \xi_{it,i(t+1)}(\ell, k)\right) \\ &= \log\left(\sum_{k=1}^{K} \exp\left\{\log \xi_{it,i(t+1)}(\ell, k)\right\}\right) \end{aligned} \tag{B.10.3}$$

and for $t = T$

$$\begin{aligned}\log \tau_{iT}(k) &= \log\left(\sum_{\ell=1}^{K} \xi_{i(T-1),iT}(\ell, k)\right) \\ &= \log\left(\sum_{\ell=1}^{K} \exp\left\{\xi_{i(T-1),iT}(\ell, k)\right\}\right) \; \forall k \in \{1, 2, \cdots .K\} \end{aligned} \tag{B.10.4}$$

All in One place:

Now that we have derived all the needed steps, here is the final steps to the implementation using $\tau_{it}(k)$ and $\xi_{it,i(t+1)}(\ell, k)$:
From Eq(B.2.4), k_o is just an index. We replace it k here.

$$\begin{aligned}\hat{\pi}_k^{(s+1)} &= \frac{\sum_{i=1}^{n} p(z_{i1} = k | x_{i1:iT}, \boldsymbol{\theta}^s)}{n} \\ &= \frac{1}{n}\sum_{i=1}^{n} \tau_{i1}(k) \; \forall k \in \{1, 2, \cdots, K\}\end{aligned} \tag{B.10.5}$$

From Eq(B.3.2), also changing indices from k_o, ℓ_o to k, ℓ We have

$$\begin{aligned}\hat{A}_{\ell k}^{(s+1)} &= \frac{\sum_{i=1}^{n}\sum_{t=1}^{T-1} p(z_{i(t+1)} = k, z_{it} = \ell | x_{i1:iT}, \boldsymbol{\theta}^s)}{\sum_{i=1}^{n}\sum_{t=1}^{T-1} p(z_{it} = \ell | x_{i1:iT}, \boldsymbol{\theta}^s)} \\ &= \frac{\sum_{i=1}^{n}\sum_{t=1}^{T-1} \xi_{it,i(t+1)}(\ell, k)}{\sum_{i=1}^{n}\sum_{t=1}^{T-1} \tau_{it}(\ell)} \; \forall \ell, k \in \{1, 2, \cdots, K\}\end{aligned} \tag{B.10.6}$$

From Eq(B.4.3), also changing indices from k_o, m_o to k, m We have:

$$\begin{aligned}\hat{B}_m^{(s+1)}(k) &= \frac{\sum_{i=1}^{n}\sum_{t=1}^{T} p(z_{it} = k | x_{i1:iT}, \boldsymbol{\theta}^s)\mathbb{I}(x_{it} = m)}{\sum_{i=1}^{n}\sum_{t=1}^{T} p(z_{it} = k | x_{i1:iT}, \boldsymbol{\theta}^s)} \\ &= \frac{\sum_{i=1}^{n}\sum_{t=1}^{T} \tau_{it}(k)\mathbb{I}(x_{it} = m)}{\sum_{i=1}^{n}\sum_{t=1}^{T} \tau_{it}(k)} \forall k \in \{1, 2, \cdots, K\}, \forall m \in \{1, 2, \cdots, M\}\end{aligned} \tag{B.10.7}$$

And we are finally ready, to implement. The key is to compute everything in log space, and then revert to probability space just before the updates.