HMM WITH MULTIDIMENSIONAL BINARY OBSERVATIONS

A.2 Advanced Signal Processing

Author:

MANUEL MONTOYA CATALÁ

1. INTRODUCTION

The aim of this assignment is to develop a Baum-Welch algorithm for an HMM which has the following specifications:

- D-dimensional observations $\overline{\boldsymbol{y}} = [y_1, \dots, y_d, \dots, y_D]$ where the emission probability associated with the state $i \in I$, $i = 1, \dots, I$ is defined as:

$$P_i(\overline{\boldsymbol{y}}|\overline{\boldsymbol{b}}_{\boldsymbol{i}}) = \prod_{d=1}^{D} b_{id}^{y_d}(1 - b_{id})^{1-y_d} = \begin{bmatrix} P(\overline{\boldsymbol{y}} = \overline{\boldsymbol{y}}_{\boldsymbol{i1}}) \\ P(\overline{\boldsymbol{y}} = \overline{\boldsymbol{y}}_{\boldsymbol{i2}}) \\ \vdots \\ P(\overline{\boldsymbol{y}} = \overline{\boldsymbol{y}}_{\boldsymbol{i}|\overline{\boldsymbol{y}}_i|}) \end{bmatrix}$$

Where the cardinality of $\overline{\boldsymbol{y}}_{\boldsymbol{i}}$ is $|\overline{\boldsymbol{y}}_i| = 2^D$ and it is the same for every state $i \in I$

We have N realizations of the HMM $\{\boldsymbol{Y^n}, \boldsymbol{S^n}\}$, with $\boldsymbol{Y} = \{\boldsymbol{Y^1}, \boldsymbol{Y^2}, \dots, \boldsymbol{Y^N}\}$, $\boldsymbol{S} = \{\boldsymbol{S^1}, \boldsymbol{S^2}, \dots, \boldsymbol{S^N}\}$ each realization has T time index elements, $\boldsymbol{Y^n} = \overline{\boldsymbol{y}}_{\boldsymbol{1:T}}^{\boldsymbol{n}} = \{\overline{\boldsymbol{y}}_{\boldsymbol{1}}^{\boldsymbol{n}}, \overline{\boldsymbol{y}}_{\boldsymbol{2}}^{\boldsymbol{n}}, \dots, \overline{\boldsymbol{y}}_{\boldsymbol{T}}^{\boldsymbol{n}}\}$, $\boldsymbol{S^n} = \boldsymbol{s}_{\boldsymbol{1:T}}^{\boldsymbol{n}} = \{s_1^n, s_2^n, \dots, s_1^n\}$.

The complete data likelihood of one of these realizations $\{\boldsymbol{Y^n}, \boldsymbol{S^n}\}$ as:

$$I_C(\boldsymbol{\theta}) = p(Y, S|\boldsymbol{\theta}) = \underbrace{\left(p(s_1|\boldsymbol{\theta}) \prod_{t=2}^{T} p(s_t|s_{t-1}, \boldsymbol{A})\right)}_{p(S|\overline{\boldsymbol{\theta}})} \cdot \underbrace{\left(\prod_{t=1}^{T} p(\overline{\boldsymbol{y}}_{\boldsymbol{t}}|s_t, \boldsymbol{B})\right)}_{p(Y|S,\overline{\boldsymbol{\theta}})}$$

In this paper we will:

- Run the HMM algorithm for obtaining the parameters of the model $\boldsymbol{\theta} = \{\overline{\boldsymbol{\pi}}, \boldsymbol{A}, \boldsymbol{B}\}$ for different number of states I and pick one of them
- Decode the given realizations $\boldsymbol{Y} = \{\boldsymbol{Y^1}, \boldsymbol{Y^2}, \dots, \boldsymbol{Y^N}\}$ using:
  - Step-by-Step MAP decoder
  - ML decoder with Viterbi
  - MAP decoder with Viterbi

We will derive the different equations needed for the algorithm, code the algorithm in MATLAB and use it on 2 the data set provided.
This report consists of 2 parts:

1) Work Description: Brief answers to the assignment questions based on the theoretical background.
2) Theoretical Background: All the theory needed to do the assignment. Actually, all the hard work is made here.

All the codes for performing the HMM and the decoders is provided.

2. WORK DESCRIPTION

In this Section of the report, we will answer the questions of the assignment taking into account the Theoretical Background explained in Section **Theory background**.

2.1 Complete data log likelihood of the model

1. Write down the expression for the complete data log likelihood for N sequences

$$\log p(S, Y|\boldsymbol{\theta}) = \log \prod_{n=1}^{N} \left(p(s_1^n|\boldsymbol{\pi}) \prod_{t=2}^{T_n} p(s_t^n|s_{t-1}^n, \mathbf{A}) \right) \left(\prod_{t=1}^{T_n} p(\mathbf{y}_t^n|s_t^n, \mathbf{B}) \right)$$

where $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$ are the model parameters.

Having N realizations of the HMMs $\{\boldsymbol{Y^n}, \boldsymbol{S^n}\}$, with $\boldsymbol{Y} = \{\boldsymbol{Y^1}, \boldsymbol{Y^2}, \dots, \boldsymbol{Y^N}\}$, $\boldsymbol{S} = \{\boldsymbol{S^1}, \boldsymbol{S^2}, \dots, \boldsymbol{S^N}\}$ each realization has T time index elements, $\boldsymbol{Y^n} = \bar{\boldsymbol{y}}_{\boldsymbol{1:T}}^{\boldsymbol{n}} = \{\bar{\boldsymbol{y}}_{\boldsymbol{1}}^{\boldsymbol{n}}, \bar{\boldsymbol{y}}_{\boldsymbol{2}}^{\boldsymbol{n}}, \dots, \bar{\boldsymbol{y}}_{\boldsymbol{T}}^{\boldsymbol{n}}\}$, $\boldsymbol{S^n} = \boldsymbol{s}_{\boldsymbol{1:T}}^{\boldsymbol{n}} = \{s_1^n, s_2^n, \dots, s_1^n\}$.

As seen in Section **3.6.1** we can obtain the complete data log likelihood of one of these realizations $\{\boldsymbol{Y^n}, \boldsymbol{S^n}\}$ as:

$$\ln\big(I_C(\boldsymbol{\theta})\big) = \ln\big(p(Y, S|\boldsymbol{\theta})\big) = \ln\left(\underbrace{\left(p(s_1|\boldsymbol{\theta}) \prod_{t=2}^{T} p(s_t|s_{t-1}, \boldsymbol{A}) \right)}_{p(S|\bar{\boldsymbol{\theta}})} \cdot \underbrace{\left(\prod_{t=1}^{T} p(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|s_t, \boldsymbol{B}) \right)}_{p(Y|S, \bar{\boldsymbol{\theta}})} \right)$$

For a group N realizations of the HMM we have, that the complete log likelihood of $\{Y, S\}$ is:

$$\begin{aligned} I_C(\bar{\boldsymbol{\theta}}) = \ln\left(p(Y, S|\bar{\boldsymbol{\theta}})\right) &= \sum_{n=1}^{N} \ln\big(p(Y = Y^n, S|\boldsymbol{\theta})\big) = \\ &= \sum_{n=1}^{N} \sum_{i=1}^{I} \mathbb{I}(s_1^n = i) \underbrace{\ln\big(p(s_1^n = i|\boldsymbol{\theta})\big)}_{\ln(\pi_i)} \\ &+ \sum_{n=1}^{N} \sum_{t=2}^{T} \left(\sum_{i=1}^{I} \sum_{j=1}^{I} \mathbb{I}(s_t^n = j, s_{t-1}^n = i\ ) \underbrace{\ln\big(p(s_t^n = j|s_{t-1}^n = i, \boldsymbol{A})\big)}_{\ln(a_{ij})} \right) \\ &+ \sum_{n=1}^{N} \sum_{t=1}^{T} \left(\sum_{i=1}^{I} \mathbb{I}(s_t^n = i) \underbrace{\ln\big(p(\bar{\boldsymbol{y}}_{\boldsymbol{t}} = \bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|s_t^n = i, \boldsymbol{B})\big)}_{\ln\left(p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|\bar{\boldsymbol{b}}_{\boldsymbol{i}})\right)} \right) \end{aligned}$$

We finally obtain:

$$\begin{aligned} I_C(\boldsymbol{\theta}) = &\sum_{i=1}^{I} \left(\sum_{n=1}^{N} \mathbb{I}(s_1^n = i) \ln(\pi_i) \right) + \sum_{i=1}^{I} \sum_{j=1}^{I} \left(\sum_{n=1}^{N} \sum_{t=2}^{T} \mathbb{I}(s_t^n = j, s_{t-1}^n = i\ ) \ln(a_{ij}) \right) \\ &+ \sum_{i=1}^{I} \left(\sum_{n=1}^{N} \sum_{t=1}^{T} \mathbb{I}(s_t^n = i) \ln\left(p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|\bar{\boldsymbol{b}}_{\boldsymbol{i}})\right) \right) \end{aligned}$$

2.2 Expected complete data log likelihood of the model

2. Write down the expression for the expected complete data log likelihood

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}) = E\{l_c(\boldsymbol{\theta})|\mathcal{D}, \boldsymbol{\theta}^{t-1}\}$$

As seen in Section **3.6.2** we can obtain the expected complete data log likelihood as:

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta^{r-1}}) &= E\{I_C(\boldsymbol{\theta})|Y, \boldsymbol{\theta^{r-1}}\} \\ &= \sum_{i=1}^{I}\left(\sum_{n=1}^{N}\gamma_1^n(i)\ln(\pi_i)\right) + \sum_{i=1}^{I}\sum_{j=1}^{I}\left(\sum_{n=1}^{N}\sum_{t=2}^{T}\xi_{t-1}^n(i,j)\ln(a_{ij})\right) \\ &+ \sum_{i=1}^{I}\left(\sum_{n=1}^{N}\sum_{t=1}^{T}\gamma_t^n(i)\ln\left(p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|\bar{\boldsymbol{b}}_{\boldsymbol{i}})\right)\right) \end{aligned}$$

Where:

$$\begin{aligned} \gamma_1^n(i) &= P(s_1^n = i|\boldsymbol{Y}, \boldsymbol{\theta^{r-1}}) \\ \gamma_t^n(i) &= P(s_t^n = i|\boldsymbol{Y}, \boldsymbol{\theta^{r-1}}) \\ \xi_{t-1}^n(i,j) &= P(s_t^n = j, s_{t-1}^n = i|\boldsymbol{Y}, \bar{\boldsymbol{\theta}}^{\boldsymbol{r-1}}) \end{aligned}$$

Where:

$$\begin{aligned} \gamma_t^n(i) &\propto \alpha_t^n(i)\cdot\beta_t^n(i) \\ \xi_t^n(i,j) &\propto \alpha_t^n(i)a_{ij}p_j(\bar{\boldsymbol{y}}_{\boldsymbol{t+1}}^{\boldsymbol{n}}|\bar{b}_j)\cdot\beta_{t+1}^n(j) \end{aligned}$$

Normalize using the properties:

$$\sum_{j=1}^{I}\gamma_t^n(i) = 1 \quad and \quad \sum_{i=1}^{I}\sum_{j=1}^{I}\xi_t^n(i,j) = 1$$

The alphas and betas are obtained as:

Formula for the alphas:

$$\begin{aligned} &\alpha_1^n(i) = \pi_i\cdot p_i(\bar{y}_1^n|\,\bar{b}_i) && i = 1,\dots,I \quad n = 1,\dots,N \\ &\alpha_t^n(i) = p_i(\bar{y}_t^n|\,\bar{b}_i)\sum_{j=1}^{I}a_{ji}\alpha_{t-i}^n(j) && i = 1,\dots,I \quad t = 2,\dots,T \quad n = 1,\dots,N \end{aligned}$$

Formula for the betas:

$$\begin{aligned} &\beta_T^n(i) = 1 && i = 1,\dots,I \quad n = 1,\dots,N \\ &\beta_t^n(i) = \sum_{j=1}^{I}a_{ij}\cdot p_j(\bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|\,\bar{\boldsymbol{b}}_{\boldsymbol{j}})\cdot\beta_{t+1}^n(j) && i = 1,\dots,I \quad t = 1,\dots,T-1 \quad n = 1,\dots,N \end{aligned}$$

2.3 Expression for the ML estimates

3. Derive the expression of the ML estimates of the new set of parameters $\boldsymbol{\theta}^t$

$$\boldsymbol{\theta}^t = \underset{\theta}{\operatorname{argmax}}\, Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1})$$

As seen in Section **3.6.1** we can obtain the ML estimates of the new set of parameters $\boldsymbol{\theta^r} = \{\bar{\boldsymbol{\pi}}, \boldsymbol{A}, \boldsymbol{B}\}^r$ derivating $Q(\boldsymbol{\theta}, \boldsymbol{\theta^{r-1}})$ with respect to $\boldsymbol{\theta}$ and equate to 0.

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta^{r-1}}) = \underbrace{\sum_{i=1}^{I}\left(\sum_{n=1}^{N}\gamma_1^n(i)\ln(\pi_i)\right)}_{Depends\ on\ \bar{\pi}} + \underbrace{\sum_{i=1}^{I}\sum_{j=1}^{I}\left(\sum_{n=1}^{N}\sum_{t=2}^{T}\xi_{t-1}^n(i,j)\ln(a_{ij})\right)}_{Depends\ on\ A} + \underbrace{\sum_{i=1}^{I}\left(\sum_{n=1}^{N}\sum_{t=1}^{T}\gamma_t^n(i)\ln(p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|b_i))\right)}_{Depends\ on\ B}$$

- **Probabilities of the initial state $\bar{\boldsymbol{\pi}}$**

We have that:

$$\pi_i = \frac{N_i}{N} = \frac{1}{N}\sum_{n=1}^{N}\gamma_1^n(i) \qquad\qquad N = \sum_{i=1}^{I} N_i$$

- **Transition probabilities $\boldsymbol{A}$**

We have that:

$$a_{ij} = \frac{\boldsymbol{E_{ij}}}{\boldsymbol{E_i}} = \frac{\boldsymbol{1}}{\boldsymbol{E}}\sum_{n=1}^{N}\sum_{t=2}^{T}\xi_{t-1}^n(i,j) \qquad\qquad \boldsymbol{E_i} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{I}}\sum_{n=1}^{N}\sum_{t=2}^{T}\xi_{t-1}^n(i,j)$$

- **Values of the parameters of the random distributions b**

Since the observation emission probabilities $P_i(\bar{\boldsymbol{y}}|\bar{\boldsymbol{b}}_{\boldsymbol{i}})$, follow a D-dimensional multinomial, we use the property of moment matching for the exponential family and we have:

$$E\{\phi(\bar{Y}_i)\} = \frac{1}{\Gamma_i}\sum_{n=1}^{N}\sum_{t=1}^{T}\gamma_t^n(i)\left(\phi(\bar{y}_t^n)\right) \;\; with\;\; \Gamma_i = \sum_{n=1}^{N}\sum_{t=1}^{T}\gamma_t^n(i)$$

The $d-th$ component of the $i-th$ state is:

$$b_{id} = \frac{1}{\boldsymbol{\Gamma_i}}\sum_{n=1}^{N}\sum_{t=1}^{T}\gamma_t^n(i)(\boldsymbol{y}_{td}^n)$$

2.4 Baum-Welch algorithm implementation

4. Implement the Baum-Welch algorithm for this HMM using Matlab code. The algorithm should take as input, at least, the number of hidden states, I, a cell of matrices containing the data set, the minimum increment in the log likelihood for convergence, and the maximum number of iterations. Hand in code and a high level explanation of what you algorithm does. **(This part can be done in groups)**

In Section **3.7** we explain in a very high detail how to compute the BW algorithm.
We have the following codes:

- `function [pi,A,B,logl] = HMM(I,data,delta,R)`
  This function calculates the parameters $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}\}$ of the HMM. It outputs the log-likelihood of every iteration, as well as model the model parameters `[pi,A,B]`.You have to indicate the number of states **I**, the maximum number of iterations **R** and the minimum convergence value **delta**.
- `function [best_pi,best_A,best_B,best_logl] = run_several_HMM(I,data,delta,R,N)`
  It executes the previous function **N** times, and outputs the best realization of them.

2.5 Decoders implementation

5. Implement a state-by-state MAP decoder based on the Forward-Backward algorithm and a ML sequence decoder based on the Viterbi algorithm. **(This part can be done in groups)**

In Sections **3.4** and **3.5** we describe step by step how to derive and perform the decodification of a realization of the HMM for:

- State-by-State MAP decoder
- ML decoder based on Viterbi's algorithm
- MAP decoder based on Viterbi's algorithm

We have the following codes:

- `function [S_MAP_SbS,S_ML_Vit,S_MAP_Vit] = Decoding(data, A, B, pi)`
  It decodes the sequences in data using the model `[pi,A,B]` and output the state sequences of the 3 algorithms in `[S_MAP_SbS,S_ML_Vit,S_MAP_Vit]`.

For using the different functions we have the codes:

- Example_Use.m: It contains a simple example that uses the HMM function to get the model parameters and does the decodification as well.
- Testing.m: It runs the HMM for different number of states I several times and plots the log-likelihood values in a graph altogether.

2.6 Results over the data

6. Run your algorithm on the data set for varying $I = 2, 3, 4, 5$. Verify that the log likelihood increases at each step of EM. Report the log likelihoods obtained and display the parameters and the hidden sequences found by the state-by-state MAP decoder and the ML sequence decoder. Comment the performances of the algorithms for finding the hidden sequences.

We have a set of data with the following characteristics:

- $N = 10$: Realizations of the HHM.
- Each realization has $T = 100$ observations time index vales.
- Each observation is a D-dimensional Bernoulli with $D = 6$

2.6.1 Parameter Estimation

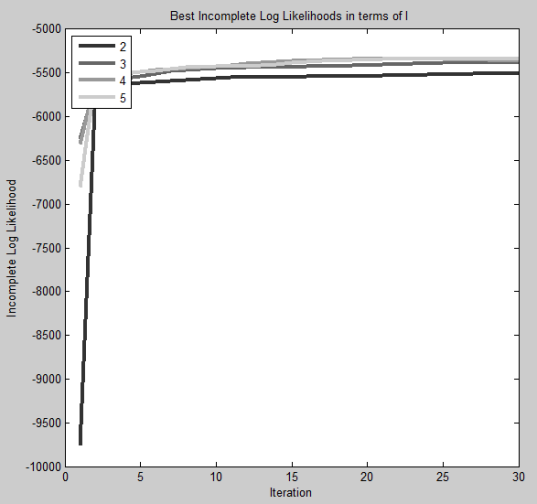
What we have done is running the HMM algorithm to get the parameters of the model

$$\boldsymbol{\theta} = \{\bar{\boldsymbol{\pi}}, \boldsymbol{A}, \boldsymbol{B}\}$$

Since we don't know what the number of states a priori is, what we do is:

- Run the HMM algorithm several times for different number of states $I = 2,3,4,5$
- For every number of states I, we pick the realization which provided us the best incomplete log-likelihood.
- Plot the best incomplete log-likelihood for every given number of states.

(Log-likelihood are calculated using base 2)



As we can see from the graph:

- The incomplete log-likelihood is always increasing for all number of states I. Notice that the incomplete log-likelihood could decrease sometimes, since the EM algorithm assures that the **complete** log-likelihood increases all the time, not the incomplete one.
- As we increase the number of states, the incomplete log-likelihood increases in general.
- There is not a great increase in the log-likelihood as we increase the number of states above 3. This could mean that 3 is the number of states of the HMM

2.6.2 Selecting number of states

We could also guess the number of parameters by looking at the $\bar{\pi}$ and $\boldsymbol{A}$ if they tell us that the probability of going and staying in a given state is very low.

2.6.2.1 Probabilities of the initial state

The following images tells us the $\bar{\pi}$ for $I = 2,3,4,5$

0.4809
0.5191

0.5485
0.1072
0.3443

0.4846
0.0850
0.3672
0.0632

0.1254
0.3158
0.0013
0.0073
0.5502

If we had a bigger number of realizations of the HMM, these values could be a good indicator of the number of states since maybe a state that has a very low initial probability is just a state modeling outliers or if 2 of them are very close together then it could be

2.6.2.2 Transition probabilities

The following images tells us the A for $I = 2,3,4,5$

0.6565	0.3435
0.3047	0.6953

0.5387	0.2029	0.2584
0.1527	0.5864	0.2608
0.1878	0.1768	0.6354

0.4828	0.2943	0.0061	0.2168
0.1690	0.2529	0.4117	0.1663
0.2000	0.4676	0.1340	0.1984
0.1501	0.2235	0.0333	0.5931

0.6114	0.0390	0.0945	0.0935	0.1616
0.2061	0.3596	0.1611	0.0757	0.1976
0.2122	0.4622	0.0014	0.0758	0.2483
0.1867	0.3351	0.3466	0.0114	0.1202
0.2344	0.0658	0.0466	0.1195	0.5337

- Each vector i tells us what is the probability of going to the state i from any state j.
- Each row j tells us what is the probability of going to any state i from state j

If all the values of a vector i are very low compared with the other states, it means that it is very unlikely to fall into that state in the HMM which could mean that that state does not actually exists.

2.6.2.1 Values of the parameters of the random distributions

The following images tells us the B for $I = 2,3,4,5$.
Each row identifies the parameters associated to each state of the HMM.

1	0.3919	0.5883	0.3112	0.8948	0.1814	0.5156
2	0.6455	0.6955	0.2579	0.0708	0.3924	0.2479

1	0.8622	0.5291	0.1311	0.1035	0.1220	0.0115
2	0.4490	0.9007	0.3660	0.2796	0.7866	0.4571
3	0.3627	0.5317	0.3211	0.8247	0.0397	0.5501

1	0.8409	0.5228	0.1389	0.0045	0.1081	0.0016
2	0.3347	0.4228	0.3440	0.8626	0.0724	0.5113
3	0.5593	0.7351	0.1928	0.9055	0.0942	0.5183
4	0.4410	0.8979	0.3821	0.2274	0.7319	0.4669

1	0.4264	0.8494	0.3829	0.2987	0.7249	0.4692
2	0.5028	0.6941	0.2069	0.8422	7.8493e-05	0.5892
3	0.4620	0.7286	0.2644	1.0000	7.4682e-11	0.5024
4	0.1001	0.0269	0.5006	0.8657	0.0276	0.5481
5	0.8297	0.5099	0.1400	0.0845	0.1151	0.0082

Taking all of these data into account we choose $I = 3$ to be the number of states of the HMM

2.6.3 Sequences Decodification

Now we will show the decodification sequences of the 10 realizations of the HMM, but just for the first 12 states, because the images are very big.

2.6.3.1 State-by-State MAP decoder

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	2	1	3	1	2	1	2	2	2	2
2	1	1	2	2	2	2	3	3	3	3	1	3
3	2	2	2	3	1	2	1	1	1	1	2	3
4	1	3	3	1	1	3	1	1	1	3	3	2
5	1	3	3	3	3	3	1	3	1	3	3	3
6	3	3	2	3	2	2	3	3	3	3	1	1
7	1	1	1	3	1	1	1	1	3	2	3	3
8	1	2	2	2	3	1	3	3	1	2	2	2
9	3	3	1	1	1	1	1	1	1	1	3	3
10	3	3	3	2	3	3	2	1	1	1	1	1

2.6.3.2 ML decoder based on Viterbi's algorithm

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	2	1	3	1	2	1	2	1	2	2
2	1	1	2	2	2	2	3	3	3	3	1	3
3	2	2	2	3	1	2	1	2	1	1	2	3
4	1	3	3	1	1	3	1	1	1	3	3	2
5	1	3	3	3	3	3	1	2	1	2	3	3
6	3	3	2	3	2	2	3	3	2	3	1	2
7	1	1	1	3	1	1	3	1	3	2	3	3
8	3	2	2	2	3	1	3	3	1	2	2	2
9	3	3	1	1	1	1	1	1	1	1	3	3
10	3	3	3	2	3	3	2	1	1	3	1	1
11												

2.6.3.3 MAP decoder based on Viterbi's algorithm

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	2	1	3	1	1	1	2	2	2	2
2	1	1	2	2	2	2	3	3	3	3	1	3
3	2	2	2	3	1	2	1	1	1	1	2	3
4	1	3	3	1	1	3	1	1	1	3	3	2
5	1	3	3	3	3	3	1	1	1	3	3	3
6	3	3	2	2	2	2	3	3	3	3	1	1
7	1	1	1	1	1	1	1	1	3	2	3	3
8	1	2	2	2	3	1	1	1	1	2	2	2
9	3	3	1	1	1	1	1	1	1	1	3	3
10	3	3	3	2	3	3	2	1	1	1	1	1

All the decoders output almost the same state sequences which is an indicator that they work correctly and it means the problem is stable. Every one of them has different properties but the MAP based on the Viterbi is the one that minimizes the probability of error so we stick to that one.

About the time performance of the algorithms:

- Viterbi's algorithm has a complexity of $O(I^2 \cdot T)$
- State-by-State MAP decoder has a complexity of $O(I^2 \cdot T)$ due to the forward-backward algorithm.

For small values of I, the Viterbi's algorithm performs better in time, but as we increase it, they converge.

3. THEORY BACKGROUND

3.1 Discrete Markov Models

A Source is said to follow a Discrete Markov Model if:

- It outputs a sequence of discrete values $S = \{s_1, s_2, \dots, s_T\}$ from a given discrete set $I = \{1,2, \dots, I\}$. This means that $s_t \in I$.
- The components s_t of the sequence are drawn in discrete time intervals $t = 1,2, \dots, T$
- The probability of drawing a given value i at time t, it only depends on the state at time $t-1$.

$$P(s_t = i|s_{1:t-1}) = P(s_t = i|s_{t-1}, s_{t-2}, \dots, s_1) = P(s_t = i|s_{t-1})$$

This model is defined by:

- The initial state probability

$$P(s_1) = \bar{\boldsymbol{\pi}} = [\pi_1, \pi_2, \dots, \pi_I]$$

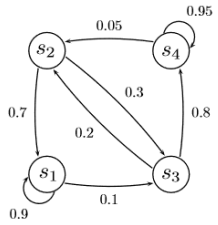
This is the probability of every state to be the first state drawn from the HMM.

- Transition probabilities:
  p_{ij}: Is the probability of going to the state j, from state i.

$$p_{ij} = P(s_t = j|s_{t-1} = i) \quad With \quad \sum_{j=1}^{I} p_{ij} = 1$$

These probabilities form the transition matrix:

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1I} \\ p_{21} & p_{22} & \cdots & p_{2I} \\ \vdots & \vdots & \ddots & \vdots \\ p_{I1} & p_{I2} & \cdots & p_{II} \end{pmatrix} \quad Rows\ sum\ 1$$



By the chain Rule and the property of the Markov Models we have:

$$\mathbf{P} = \begin{pmatrix} 0.9 & 0 & 0.1 & 0 \\ 0.7 & 0 & 0.3 & 0 \\ 0 & 0.2 & 0 & 0.8 \\ 0 & 0.05 & 0 & 0.95 \end{pmatrix}$$

$$P(s_t = i|s_{t-2} = j) = \frac{P(s_t = i, s_{t-2} = j)}{P(s_{t-2} = j)} = \frac{\sum_{k=1}^{I} P(s_t = i, s_{t-1} = k, s_{t-2} = j)}{P(s_{t-2} = j)}$$
$$= \frac{\sum_{k=1}^{I} P(s_{t-2} = j) \cdot P\big(s_{t-1} = k|P(s_{t-2} = j)\big) \cdot P\big(s_t = i|P(s_{t-1} = k)\big)}{P(s_{t-2} = j)}$$
$$= \sum_{k=1}^{I} P\big(s_{t-1} = k|P(s_{t-2} = j)\big) \cdot P\big(s_t = 1|P(s_{t-1} = k)\big) = \sum_{k=1}^{I} p_{ik} p_{kj}$$

We have that the probability of being at state i when we know that 2 time index before we are at state j is:

$$P(s_t = i|s_{t-2} = j) = P_{ij}^2$$

Appying this we have that, $P(s_t = i) = \boldsymbol{P^{t-1}} \cdot \bar{\boldsymbol{\pi}}$

3.2 Hidden Discrete Markov Models

Am HMM is just a Discrete Markov Model in which we cannot see the state s_t drawn, instead, we can see a noise version of this state y_t given by the emission probability functions of each state.

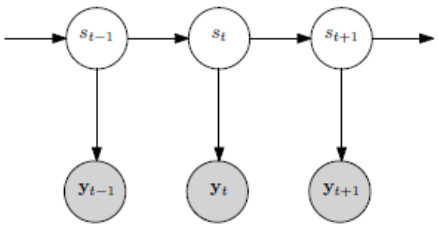
$$p_i(\bar{y}|\bar{b}_i)$$

This is why the states are called hidden states.

The problem is that, in general, any state $s_t \in I$ could be observed as any output value $y_t \in R$, so, looking at the observed value, we cannot be sure to which state it belongs to, it could be any of them, but some of them will be more likely than others.

So we have probability mass functions$p_i(\bar{y}|\bar{b}_i)$ that tells us the probability of drawing the value y_t from the distribution k.

We can summarize the model as:



- $S = \{s_1, s_2, \ldots, s_T : s_t \in 1, \ldots, I\}$: hidden state sequence
- $Y = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T : \mathbf{y}_t \in \mathbb{R}^M\}$: observed continuous sequence
- $\mathbf{A} = \{a_{ij} : a_{ij} = p(s_{t+1} = j | s_t = i)\}$: state transition probabilities.
- $\mathbf{B} = \{b_i : p_{b_i}(\mathbf{y}_t) = p(\mathbf{y}_t | s_t = i)\}$: observation emission probabilities.
- $\boldsymbol{\pi} = \{\pi_i : \pi_i = p(s_1 = i)\}$: initial state probability distribution.
- $\theta = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$: model parameters.

- **The initial state probabililies:**

This is the probability of every state to be the first state drawn from the HMM.

$$\bar{\boldsymbol{\pi}} = P(s_1) = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_I \end{bmatrix}$$

- **The state transition probabilities:**

These are the probabilities of going to the state j, from state i.

$$A = P = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1I} \\ a_{21} & a_{22} & \dots & a_{2I} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1} & a_{I2} & \dots & a_{II} \end{pmatrix} \quad a_{ij} = P(s_t = j | s_{t-1} = i) \qquad Rows\ Sum\ 1$$

a_{ij}: Probability of going to state j from state i.

- **The observation emission probabilities:**

Every state $i \in I$ will have a different observation emission probability $p_i(\bar{\boldsymbol{y}}|\bar{b}_i)$.
$p_i(\bar{\boldsymbol{y}}|\bar{b}_i)$ is the probability density of observing $\bar{\boldsymbol{y}}$ when we are at state i.

$$p_i(\bar{\boldsymbol{y}}_t|\bar{b}_i) = p(\bar{\boldsymbol{y}}_t|s_t = i, \bar{b}_i)$$

$\bar{b}_i = [b_1, \dots, b_d, \dots, b_D]$ is the vector of parameters of the distribution

$$\begin{bmatrix} p_1(\bar{\boldsymbol{y}}|\bar{b}_1) \\ p_2(\bar{\boldsymbol{y}}|\bar{b}_2) \\ \vdots \\ p_I(\bar{\boldsymbol{y}}|\bar{b}_I) \end{bmatrix} = \begin{bmatrix} p(\bar{\boldsymbol{y}}_t|s_t = 1, \bar{b}_1) \\ p(\bar{\boldsymbol{y}}_t|s_t = 2, \bar{b}_2) \\ \vdots \\ p(\bar{\boldsymbol{y}}_t|s_t = I, \bar{b}_I) \end{bmatrix} \quad with\ matrix\ \boldsymbol{B} = \begin{bmatrix} \bar{\boldsymbol{b}}_{\boldsymbol{1}} \\ \bar{\boldsymbol{b}}_{\boldsymbol{2}} \\ \vdots \\ \bar{\boldsymbol{b}}_{\boldsymbol{I}} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1D} \\ b_{21} & b_{22} & \cdots & b_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ b_{I1} & b_{I2} & \cdots & b_{ID} \end{bmatrix}$$

If all the distributions have the same number of parameters D, then the observation emission probabilities matrix has $I \cdot D$ parameters

3.2.1 Case of discrete emission probabilities

If the observation emission probabilities are discrete, that means that the observations $\bar{\boldsymbol{y}}$ are discrete values and so we have that the emission probabilities can be expressed as a vector:

$$P_i(\bar{\boldsymbol{y}}|\bar{b}_i) = P_i(\bar{\boldsymbol{y}}|s_t = i, \bar{b}_i) = \begin{bmatrix} P_i(\bar{\boldsymbol{y}} = \bar{\boldsymbol{y}}_1|\bar{b}_1) \\ P_i(\bar{\boldsymbol{y}} = \bar{\boldsymbol{y}}_2|\bar{b}_1) \\ \vdots \\ P_i(\bar{\boldsymbol{y}} = \bar{\boldsymbol{y}}_O|\bar{b}_1) \end{bmatrix}$$

The number of possible values $\bar{\boldsymbol{y}}$ can take, denoted as O, is defined by $P_i(\bar{\boldsymbol{y}}|\bar{b}_i)$ and it does not depend on the number of states or anything.

This fact will not change much the mathematical derivation of the formulas, just replace $p_i(\bar{\boldsymbol{y}}_t|\bar{b}_i)$ by $P_i(\bar{\boldsymbol{y}}_t|\bar{b}_i)$ everywhere.

When we talk about likelihoods in the continuous case, we are talking about probabilities in this discrete case.

The discrete case could be seen as a continuous case where we have defined decision boundaries that gives us the different possible discrete values $\bar{\boldsymbol{y}}_O$. Ans so:

$$P_i(\overline{\boldsymbol{y_d}} = \bar{\boldsymbol{y}}_o|\bar{b}_i) = \int_O p_i(\overline{\boldsymbol{y_c}}|\bar{b}_i) d\overline{\boldsymbol{y_c}}$$

Where $\overline{\boldsymbol{y_d}}$ is the discrete R.V. and $\overline{\boldsymbol{y_c}}$ the continuous R.V.

3.2.1.1 Analogy to Digital Communications

This is just like a communication system, where:

- The states are the symbols sent through the channel.
- The $\bar{\boldsymbol{y}}_t$ values are the symbols obtained at the receiver.
- The symbols sent are not independent; the probability of sending a symbol j in the instant t depends on the symbol sent i at instant $t-1$.

3.2.1.1 Analogy to Mixture Models

A Mixture model could be modeled as an HMM where:

- The I states of the HMM are the K components of the mixture model.
- The emission probabilities $p_i(\bar{\boldsymbol{y}}_t|\bar{b}_i)$ associated with every state are the probability density functions of every component of the mixture model $p_k(\bar{\boldsymbol{y}}_t|\bar{\theta}_k)$
- The probability of being at state I at time t, does not depend on any other state.

$$P(s_t = j, s_{t-1} = i) = P(s_t = i)P(s_{t-1} = i)$$

That is, the state transition probabilities does not depend on the index i, it is like having an $\boldsymbol{A}$ with all the rows the same and also equal to $\bar{\boldsymbol{\pi}}$

3.3 Forward Backward Algorithm

Given a sequence of observations $\boldsymbol{Y} = \{\bar{\boldsymbol{y}}_{\boldsymbol{1}}, \bar{\boldsymbol{y}}_{\boldsymbol{2}}, \ldots, \bar{\boldsymbol{y}}_{\boldsymbol{T}}\}$ and the parameters $\boldsymbol{\theta} = \{\boldsymbol{A}, \boldsymbol{B}, \bar{\boldsymbol{\pi}}\}$, we want to calculate the probabily of being in state i, at time t

$$P(s_t = i|\boldsymbol{Y}, \boldsymbol{\theta})$$

The forward-backward algorithm is:
An algorithm that calculates the $P(s_t = i|\boldsymbol{Y}, \boldsymbol{\theta})$ using "Dynamic Programming" (Reuses earlier computations). It calculates, the probability that we are in state state i, at time t, given all the observations and the model.
Note that if we are at a given state s_t, its value is conditionally dependent to the one in s_{t-1} and also in s_{t+1}, so if we have that info, me must use it.

$$P(s_t = i|\boldsymbol{Y}, \boldsymbol{\theta})$$

For now on, we will omit the conditioning$|\theta$, it just means that the model is already set, so just remember that all the parameters are given.
We denote $\bar{y}_{a:b}$ as the vector of observations $= \{\bar{y}_a, \bar{y}_{a+1}, \ldots, \bar{y}_b\}$

Using Bayes we have:

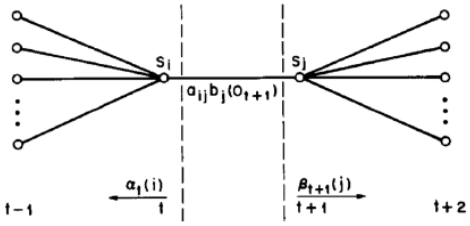
$$P(s_t = i|\boldsymbol{Y}) = P(s_t = i|\boldsymbol{Y}) = \frac{P(s_t = i, \boldsymbol{Y})}{P(\boldsymbol{Y})} \alpha \; P(s_t = k, \boldsymbol{Y})$$

We don't care about $P(Y|\theta)$ because we know that:

$$\sum_{i=1}^{I} P(s_t = i|\boldsymbol{Y}) = 1$$

So, since $P(\boldsymbol{Y}|\boldsymbol{\theta})$ does not depend on i, we can just treat it as a normalization constant.

$$P(s_t = i, \boldsymbol{Y}) = \underbrace{P(s_t = i, \bar{\boldsymbol{y}}_{\boldsymbol{1:t}})}_{Forward} \underbrace{p(\boldsymbol{y_{t+1:T}}|s_t = i)}_{Backward}$$



So we have splitted $P(s_t = i, \boldsymbol{Y})$ into two components:

- $P(s_t = i, \bar{\boldsymbol{y}}_{\boldsymbol{1:t}})$: It is the forward component. The probability of being at state i at time t, and observe the past sequences $\bar{y}_{1:t}$.
- $p(\bar{\boldsymbol{y}}_{\boldsymbol{t+1:T}}|s_t)$: It is the backward component. The probability of observing future sequences $\bar{y}_{t+1:T}$ when we have the state i at time t.

With these guys we can do:

- Arbitrary inference
- Parameter estimation
- Sampling the posterior

3.3.1 Forward Step:

In this step we wish to calculate the probability of being at state $s_t = i$ and have drawn the previous samples $\bar{\boldsymbol{y}}_{\boldsymbol{1:t}}$

$$p(\bar{\boldsymbol{y}}_{\boldsymbol{1:t}}, s_t) = Probability\ of\ having\ oberved\ \bar{y}_{1:t}\ and\ having\ last\ state\ s_t$$

We can calculate $p(\bar{\boldsymbol{y}}_{\boldsymbol{1:t}}, s_t)$ as the marginalization of $p(\bar{\boldsymbol{y}}_{\boldsymbol{1:t}}, s_t, s_{t-1})$ over the domain of I

$$p(\bar{\boldsymbol{y}}_{\boldsymbol{1:t}}, s_t) = \sum_{j=1}^{I} p(\bar{\boldsymbol{y}}_{\boldsymbol{1:t}}, s_t, s_{t-1} = j)$$

We split $\bar{\boldsymbol{y}}_{\boldsymbol{1:t}}$ into $\{\bar{\boldsymbol{y}}_{\boldsymbol{t}}, \bar{\boldsymbol{y}}_{\boldsymbol{1:t-1}}\}$ and apply chain rule:

$$p(\bar{\boldsymbol{y}}_{\boldsymbol{1:t}}, s_t, s_{t-1}) = p(\bar{\boldsymbol{y}}_{\boldsymbol{t}} \mid s_t, s_{t-1}, \bar{\boldsymbol{y}}_{\boldsymbol{1:t-1}}) P(s_t \mid s_{t-1}, \bar{\boldsymbol{y}}_{\boldsymbol{1:t-1}}) P(s_{t-1}, \bar{\boldsymbol{y}}_{\boldsymbol{1:t-1}})$$

Due to the Markov property, the terms in green are conditionally independent with the variable, so we delete them, obtaining:

$$p(\bar{\boldsymbol{y}}_{\boldsymbol{t}}, \bar{\boldsymbol{y}}_{\boldsymbol{1:t-1}}, s_t, s_{t-1}) = p(\bar{\boldsymbol{y}}_{\boldsymbol{t}} \mid s_t) P(s_t \mid s_{t-1}) P(s_{t-1}, \bar{\boldsymbol{y}}_{\boldsymbol{1:t-1}})$$

We call
$\alpha_t(i)$: Probability of being at state $s_t = i$ when we have observed the set $\bar{y}_{1:t}$

$$\alpha_t(i) = P(s_t = i, \bar{\boldsymbol{y}}_{\boldsymbol{1:t}}) = \sum_{j=1}^{I} p(\bar{\boldsymbol{y}}_{\boldsymbol{1:t}}, s_t = 1, s_{t-1} = j)$$

And as we have just seen:

$$\alpha_t(i) = \sum_{j=1}^{I} \underbrace{p(\bar{\boldsymbol{y}}_{\boldsymbol{t}} \mid s_t = i)}_{p_i(\bar{y}_t \mid \bar{b}_i)} \underbrace{P(s_t = i \mid s_{t-1} = j)}_{a_{ji}} \underbrace{P(s_{t-1} = j, \bar{\boldsymbol{y}}_{\boldsymbol{1:t-1}})}_{\alpha_{t-i}(j)}$$

So we can express $\alpha_t(i)$ in terms of the parameters of the model:

- a_{ji}: Probability of going from state j, to state i.
- $p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}} \mid \bar{\boldsymbol{b}}_{\boldsymbol{i}})$: Probability density of getting y_t from distribution i
- $\alpha_{t-i}(j)$: So we have to calculate this iteratively

We end up having the equations:

$$\alpha_1(i) = \pi_i \cdot p_i(\bar{\boldsymbol{y}}_{\boldsymbol{1}} \mid \bar{\boldsymbol{b}}_{\boldsymbol{i}}) \qquad i = 1, \dots, I$$

$$\alpha_t(i) = p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}} \mid \bar{\boldsymbol{b}}_{\boldsymbol{i}}) \sum_{j=1}^{I} a_{ji} \alpha_{t-i}(j) \qquad i = 1, \dots, I \quad t = 2, \dots, T$$

3.3.2 Backward Step

The probability of observing future sequences $\bar{\boldsymbol{y}}_{\boldsymbol{t+1:T}}$ when we have the state i at time t,

$$p(\bar{\boldsymbol{y}}_{\boldsymbol{t+1:T}}|s_t): Probability\ of\ observing\ y_{t+1:T}\ when\ we\ have\ the\ state\ i\ at\ time\ t$$

We can calculate $p(\bar{\boldsymbol{y}}_{\boldsymbol{t+1:T}}|s_t)$ as the marginalization of $p(\bar{\boldsymbol{y}}_{\boldsymbol{t+1:T}}, s_{t+1}|s_t)$over the domain of I

$$p(\bar{\boldsymbol{y}}_{\boldsymbol{t+1:T}}, s_{t+1}|s_t) = \sum_{j=1}^{I} p(\bar{\boldsymbol{y}}_{\boldsymbol{t+1:T}}, s_{t+1} = j|s_t)$$

We split $\bar{\boldsymbol{y}}_{\boldsymbol{t+1:T}}$ into $\{\bar{\boldsymbol{y}}_{\boldsymbol{t+2:T}}, \bar{\boldsymbol{y}}_{\boldsymbol{T}}\}$ and apply chain rule:

$$p(\bar{\boldsymbol{y}}_{\boldsymbol{t+2:T}}, \bar{\boldsymbol{y}}_{\boldsymbol{T}}, s_t, s_{t+1}) = p(\bar{\boldsymbol{y}}_{\boldsymbol{t+2:T}}|\ s_t, s_{t+1}, \bar{\boldsymbol{y}}_{\boldsymbol{1:t+1}}) p(\bar{\boldsymbol{y}}_{\boldsymbol{1:t+1}}|s_{t+1}, s_t) P(s_{t+1}|s_t)$$

Due to the Markov property, the terms in green are conditionally independent with the variable, so we delete them, obtaining:

$$p(\bar{\boldsymbol{y}}_{\boldsymbol{t+1:T}}, s_{t-1}|s_t) = p(\bar{\boldsymbol{y}}_{\boldsymbol{t+2:T}}|\ s_{t+1}) p(\bar{\boldsymbol{y}}_{\boldsymbol{t+1}}|s_{t+1}) P(s_{t+1}|s_t)$$

$\beta_t(i)$: Probability of being at state $s_t = i$ when we have observed the set $\bar{y}_{1:t}$

$$\beta_t(i) = p(\bar{\boldsymbol{y}}_{\boldsymbol{t+1:T}}, s_{t-1}|s_t = i) = \sum_{j=1}^{I} p(\bar{\boldsymbol{y}}_{\boldsymbol{t+1:T}}, s_{t+1} = j|s_t = i)$$

And as we have just seen:

$$\beta_t(i) = \sum_{j=1}^{I} \underbrace{p(\bar{\boldsymbol{y}}_{\boldsymbol{t+2:T}}|\ s_{t+1} = j)}_{\beta_{t+1}(j)} \underbrace{p(\bar{\boldsymbol{y}}_{\boldsymbol{t+1}}|s_{t+1} = j)}_{p_j(\bar{y}_t|\bar{\boldsymbol{b}}_{\boldsymbol{j}})} \underbrace{P(s_{t+1} = j|s_t = i)}_{a_{ij}}$$

So we can express $\beta_t(i)$ in terms of the parameters of the model:

- a_{ij}: Probability of going from state i, to state j.
- $p_i(\bar{\boldsymbol{y}}|\bar{b}_i)$: Probability density of getting $\bar{\boldsymbol{y}}$ from distribution i
- $\beta_{t+i}(j)$: So we have to calculate this iteratively.

We end up having the equations:

$$\beta_T(i) = 1 \qquad i = 1, \dots, I$$

$$\beta_t(i) = \sum_{j=1}^{I} a_{ij} \cdot p_j(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|\ \bar{\boldsymbol{b}}_{\boldsymbol{j}}) \cdot \beta_{t+1}(j) \qquad i = 1, \dots, I \quad t = 1, \dots, T-1$$

We now know how to compute the probability of being in state i at time t as:

$$P(s_t = i|\boldsymbol{Y}, \boldsymbol{\theta}) = \gamma_t(i) = \frac{\alpha_t(i) \cdot \beta_t(i)}{P(\boldsymbol{Y}|\boldsymbol{\theta})} \propto \alpha_t(i) \cdot \beta_t(i)$$

We don't care about $P(\boldsymbol{Y}|\boldsymbol{\theta})$ since we know that for a given time index t

$$\sum_{i=1}^{I} P(s_t = i|\boldsymbol{Y}, \boldsymbol{\theta}) = \sum_{i=1}^{I} \gamma_t(i) = 1$$

3.4 Likelihood of a sequence

Having N realizations of the HMMs $\{\boldsymbol{Y^n}, \boldsymbol{S^n}\}$, with $\boldsymbol{Y} = \{\boldsymbol{Y^1}, \boldsymbol{Y^2}, \dots, \boldsymbol{Y^N}\}$, $\boldsymbol{S} = \{\boldsymbol{S^1}, \boldsymbol{S^2}, \dots, \boldsymbol{S^N}\}$ each realization has T time index elements, $\boldsymbol{Y^n} = \bar{\boldsymbol{y}}^{\boldsymbol{n}}_{\boldsymbol{1:T}} = \{\bar{\boldsymbol{y}}^{\boldsymbol{n}}_{\boldsymbol{1}}, \bar{\boldsymbol{y}}^{\boldsymbol{n}}_{\boldsymbol{2}}, \dots, \bar{\boldsymbol{y}}^{\boldsymbol{n}}_{\boldsymbol{T}}\}$, $\boldsymbol{S^n} = \boldsymbol{s}^{\boldsymbol{n}}_{\boldsymbol{1:T}} = \{s^n_1, s^n_2, \dots, s^n_1\}$.

$$L(Y|\bar{\boldsymbol{\theta}}) = \ln\left(p(Y|\bar{\boldsymbol{\theta}})\right) = \sum_{n=1}^{N} \ln\left(p(Y = Y^n|\boldsymbol{\theta})\right)$$

Each realization $\boldsymbol{Y^n}$ of the HMM has probability:

$$p(\boldsymbol{Y}|\boldsymbol{\theta}) = p(\{\bar{\boldsymbol{y}}_{\boldsymbol{1}}, \bar{\boldsymbol{y}}_{\boldsymbol{2}}, \dots, \bar{\boldsymbol{y}}_{\boldsymbol{T}}\}|\boldsymbol{\theta})$$

We can calculate it as a discretization over all the domain of the possible states at every time instant:

$$p(\boldsymbol{Y} = \boldsymbol{Y^n}|\boldsymbol{\theta}) = \sum_{S} p(\boldsymbol{Y} = \boldsymbol{Y^n}, \boldsymbol{S}|\boldsymbol{\theta}) = \sum_{s_1=1}^{I} \sum_{s_2=1}^{I} \cdots \sum_{s_T=1}^{I} p(\boldsymbol{Y} = \boldsymbol{Y^n}, \{s_1, \dots, s_T\}|\boldsymbol{\theta})$$

Being:

$$p(Y, S|\bar{\boldsymbol{\theta}}) = \left(p(s_1|\bar{\boldsymbol{\theta}}) \prod_{t=2}^{T} p(s_t|s_{t-1}, \boldsymbol{A})\right) \cdot \left(\prod_{t=1}^{T} p(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|s_t, \boldsymbol{B})\right)$$

We have that:

$$p(Y = Y^n|\boldsymbol{\theta}) = \sum_{s_1=1}^{I} \sum_{s_2=1}^{I} \cdots \sum_{s_T=1}^{I} \left(p(s_1|\bar{\boldsymbol{\theta}}) \prod_{t=2}^{T} p(s_t|s_{t-1}, \boldsymbol{A})\right) \cdot \left(\prod_{t=1}^{T} p(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|s_t, \boldsymbol{B})\right)$$

This is equal to the sum $\boldsymbol{I^T}$ elements.

It can be computed in the sum of $\boldsymbol{I^2T}$ elements with the forward algorithm since:

$$p(Y = Y^n|\boldsymbol{\theta}) = \sum_{i=1}^{I} \underbrace{P(\bar{\boldsymbol{y}}^{\boldsymbol{n}}_{\boldsymbol{1:T}}, s^n_T = i|\boldsymbol{\theta})}_{Forward} = \sum_{i=1}^{I} \alpha^n_T(i)$$

3.4.1 ML decoders

A ML decoder maximizes the probability of observing $\boldsymbol{Y} = \{\overline{\boldsymbol{y}}_{\boldsymbol{1}}, \overline{\boldsymbol{y}}_{\boldsymbol{2}}, \dots, \overline{\boldsymbol{y}}_{\boldsymbol{T}}\}$ when the hidden state sequence $\boldsymbol{S} = \{s_1, s_2, \dots, s_T\}$ was drawn. In other words we want to maximize:

$$p(\boldsymbol{Y}|\boldsymbol{S}, \boldsymbol{\theta}) = \prod_{t=1}^{T} p(\overline{\boldsymbol{y}}_{\boldsymbol{t}}|s_t, \boldsymbol{B})$$

So, the optimal sequence S^* that we are looking for is:

$$\boldsymbol{S}^* = arg\max_{\boldsymbol{S}}\{p(\boldsymbol{Y}|\boldsymbol{S}, \boldsymbol{\theta})\} \qquad with \quad p(\boldsymbol{Y}|\boldsymbol{S}, \boldsymbol{\theta}) = \frac{p(\boldsymbol{S}, \boldsymbol{Y}|\boldsymbol{\theta})}{P(\boldsymbol{S}|\boldsymbol{\theta})}$$

$$p(\boldsymbol{Y}|\boldsymbol{S}, \boldsymbol{\theta}) = \left(\prod_{t=1}^{T} p(\overline{\boldsymbol{y}}_{\boldsymbol{t}}|s_t, \boldsymbol{B})\right)$$

We could calculate his probability for any possible sequence of hidden states $\boldsymbol{S} = \{s_1, s_2, \dots, s_T\}$ and pick the one with the biggest probability. This takes I^T multiplications, or we could use the Viterbi Algorithm and we could compute in $I^2 \cdot T$ multiplications.

3.4.2 MAP decoders

We want to find the hidden state sequence $\boldsymbol{S} = \{s_1, s_2, \dots, s_T\}$ that it is more likely to have generated our observations $\boldsymbol{Y} = \{\overline{\boldsymbol{y}}_{\boldsymbol{1}}, \overline{\boldsymbol{y}}_{\boldsymbol{2}}, \dots, \overline{\boldsymbol{y}}_{\boldsymbol{T}}\}$. In other words we want to maximize $p(\boldsymbol{S}|\boldsymbol{Y}, \boldsymbol{\theta})$
The optimal sequence S^* that we are looking for is:

$$\boldsymbol{S}^* = arg\max_{\boldsymbol{S}}\{p(\boldsymbol{S}|\boldsymbol{Y}, \boldsymbol{\theta})\} \qquad with \quad p(\boldsymbol{S}|\boldsymbol{Y}, \boldsymbol{\theta}) = \frac{p(\boldsymbol{S}, \boldsymbol{Y}|\boldsymbol{\theta})}{P(\boldsymbol{Y}|\boldsymbol{\theta})}$$

Maximizing over $p(\boldsymbol{S}|\boldsymbol{Y}, \boldsymbol{\theta})$ is the same as mazimizing over $p(\boldsymbol{S}, \boldsymbol{Y}, \boldsymbol{\theta})$ since $P(\boldsymbol{Y}|\boldsymbol{\theta})$ is a constant for all values of $\boldsymbol{S}$. So:

$$\boldsymbol{S}^* = arg\max_{S}\{p(\boldsymbol{S}, \boldsymbol{Y}|\boldsymbol{\theta})\}$$

$$p(\boldsymbol{S}, \boldsymbol{Y}|\boldsymbol{\theta}) = \left(p(s_1|\overline{\boldsymbol{\theta}})\prod_{t=2}^{T} p(s_t|s_{t-1}, \boldsymbol{A})\right) \cdot \left(\prod_{t=1}^{T} p(\overline{\boldsymbol{y}}_{\boldsymbol{t}}|s_t, \boldsymbol{B})\right)$$

3.4.2.1 Step-by-Step MAP Decoder

A MAP decoder is that one that finds the sequence of states $S = \{s_1, \dots, s_T\}$ that maximizes the posterior probability of observing the set $\boldsymbol{Y} = \{\overline{\boldsymbol{y}}_{\boldsymbol{1}}, \overline{\boldsymbol{y}}_{\boldsymbol{2}}, \dots, \overline{\boldsymbol{y}}_{\boldsymbol{T}}\}$.
Step-by-Step MAP Decoder is that one, that, for every state s_t maximizes the probability of being at that state given the $\boldsymbol{Y} = \{\overline{\boldsymbol{y}}_{\boldsymbol{1}}, \overline{\boldsymbol{y}}_{\boldsymbol{2}}, \dots, \overline{\boldsymbol{y}}_{\boldsymbol{T}}\}$

$$s_t^* = arg\max_{s_t}\{p(s_t, \boldsymbol{Y}|\boldsymbol{\theta})\}$$

From the forward backward algorithm we have:

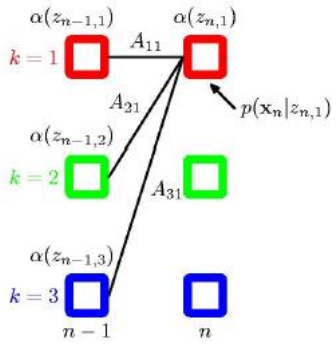
$$P(s_t = i|\boldsymbol{Y}, \boldsymbol{\theta}) = \gamma_t(i) = \frac{\alpha_t(i) \cdot \beta_t(i)}{P(\boldsymbol{Y}|\boldsymbol{\theta})} \propto \alpha_t(i) \cdot \beta_t(i)$$

So we can easily obtain:

$$s_t^* = arg\max_{s_t}\{\gamma_t(i)\}$$

3.5 Viterbi's Algorithm

Viterbi's Algorithm is just a Dynamic Programming algorithm to maximize a function $g(\cdot)$ over a set of parameters $\bar{\boldsymbol{s}}_{\boldsymbol{1:T}}$ which can expressed like:



$$g(\bar{\boldsymbol{s}}_{\boldsymbol{1:T}}) = g(s_1) \cdot g(s_2, s_1) \cdot g(s_3, s_2) \cdot\cdot g(s_T, s_{T-1})$$

Taking logarithms, we can express it as a recursive sum.

$$G(\bar{\boldsymbol{s}}_{\boldsymbol{1:T}}) = G(s_1) + G(s_2, s_1) + \cdots + G(s_T, s_{T-1})$$

Every transition between 2 states $s_{t-1} = j \rightarrow \ s_t = i$ can be modeled as a single value independent of anything else.

3.5.1 Conditions of Viterbi's Algorithm

We want to be able to express our maximization problem as $\max_{\bar{s}_{1:T}}\{G(\bar{s}_{1:T})\}$

If we have a maximization problem that we can express in the form:

$$\boldsymbol{\max_{a,b}\{f(a)\ g(a,b)\}}$$

If $f(a) \geq 0\ \forall a$ and $g(a,b) \geq 0\ \forall a, b$ then:

$$\max_{a,b}\{f(a)\ g(a,b)\} = \max_{a}\left\{ f(a) \max_{b}\{g(a,b)\}\right\}$$

And also, maximizing over $p\big(f(a)\ g(a,b)\big)$ is the same as maximizing over $\log\big(f(a)\ g(a,b)\big)$ since the function $log(\cdot)$ is monotonically increasing. So we end up with the handy expression:

$$\max_{a,b}\{f(a)\ g(a,b)\} = \max_{a,b}\{\log\big(f(a)\ g(a,b)\big)\} = \max_{a}\left\{\log(f(a)\ ) + \log\left(\max_{b}\{g(a,b)\}\right)\right\}$$
$$= \boldsymbol{\max_{a}\{F(a)\} + \max_{a}\left\{\max_{b}\{G(a,b)\}\right\}}$$

We are going to use this property to do the maximization easier and be able tu express our problem in a way that we can use Viterbi's algortithm

Our goal is to divide a huge maximization problem over a vector of values $\bar{\boldsymbol{s}}_{\boldsymbol{1:T}}$, into a set of small maximization problems recursively.

We can then use Viterbi's algorithm if what we maximize has the following property:

$$\max_{\bar{\boldsymbol{s}}_{\boldsymbol{1:t}}}\{p(\bar{\boldsymbol{s}}_{\boldsymbol{1:t}})\} = \max_{\bar{\boldsymbol{s}}_{\boldsymbol{1:t}}}\{f(s_t)p(\bar{\boldsymbol{s}}_{\boldsymbol{1:t}})\}$$

Then:

$$\max_{\bar{\boldsymbol{s}}_{\boldsymbol{1:t}}}\{P(\bar{\boldsymbol{s}}_{\boldsymbol{1:t}})\} = \max_{s_t}\{F(s_t)\} + \max_{s_t}\left\{\max_{\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}}}\{P(\bar{\boldsymbol{s}}_{\boldsymbol{1:t}})\}\right\}$$

3.5.2 Implementation of Viterbi's Algorithm

Since we want to maximize a function $G(\bar{\boldsymbol{s}}_{\boldsymbol{1:T}})$ with the property:

$$G(\bar{\boldsymbol{s}}_{\boldsymbol{1:T}}) = G(s_1) + G(s_2,s_1) + \cdots + G(s_T,s_{T-1})$$

Let's define the following variables:

- Survival_Paths(T,I): At each step t, it will contain for every row $i \in I$, the survival path that ends in the state i. Please notice that in every iteration of t, the values of *Survival_Paths(1:t-1,i)* will change if the survival path does not end in the same state as at time $t-1$.
- Past_Survival_Paths(T,I): Auxiliary variable equal to the previous one to avoid the problem of overwriting paths.
- G (T,I): At each step t, it will contain for every row $i \in I$, the value of $G(\bar{\boldsymbol{s}}_{\boldsymbol{1:t}})$ of the survival path that ends in the state i.
- G_possible (T,I): Auxiliary variable to store all the possible $G(s_t = i, s_{t-1} = j)$ for $j \in I$ in order to select the $\max_j\{G(s_t = i, s_{t-1} = j)\}$

Computation:

In the first step:

- Just calculate $G(s_1)$ for all $s_1 \in I$, updating the data matrices.

```
for i = 1:I
     Survival_Paths(1,i) = i;
     G(1,i) = G(s_1);
end
Past_Survival_Paths = Survival_Paths;
```

In the second step:

- Calculate for every possible state of $s_t = i$, that is, for every $i \in I$, what is the survival path that ends in state $s_{t-1} = j$ that gives the maximum $G(s_t = i, \bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}} = j)$

```
for t = 2:T
    for i = 1:I
         for j = 1:I
            G_Possible(j) = G(t-j,i) + G(s_t = i, s_{t-1} = j);
         end
         [Best_G_ji, best_j] = max (G_Possible (:));

         Survival_Paths(1:t-1,i) = Past_ Best_G_ji (1:t-1,best_j);
         Survival_Paths (t,i) = i;
         G(t,i) = Best_G_ji;
    end
    Past_Survival_Paths = Survival_Paths;
end
```

We select the best Survival path at step T as:

```
[G_best, best_S] = max (G(T,:));
Best_S(:) = Survival_Paths (:,best_S);
```

3.5.3 Viterbi's Algorithm for ML decoder

The maximization problem we have is:

$$\max_{\boldsymbol{S}}\{p(\boldsymbol{Y}|\boldsymbol{S},\boldsymbol{\theta})\} = \max_{\boldsymbol{s_{1:T}}}\{p(\bar{\boldsymbol{y}}_{\boldsymbol{1:T}}|\boldsymbol{s_{1:T}},\boldsymbol{\theta})\}$$

If for any index t, we can express $p(\bar{\boldsymbol{y}}_{\boldsymbol{1:T}}|\boldsymbol{s_{1:t}},\boldsymbol{\theta})$ *as*:

$$m(\boldsymbol{s_{1:t}}) = \underbrace{p(\bar{\boldsymbol{y}}_{\boldsymbol{1:T}}|\boldsymbol{s_{1:t}},\boldsymbol{\theta})}_{m(\boldsymbol{s_{1:t}})} = f(s_t)g(s_t,\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}})$$

$$m(\boldsymbol{s_{1:T}}) = \{p(\boldsymbol{Y}|\boldsymbol{S},\boldsymbol{\theta})\}$$

In ML case, we have, for a given value of t:

$$p(\bar{\boldsymbol{y}}_{\boldsymbol{1:T}}|\boldsymbol{s_{1:t}},\boldsymbol{\theta}) = \prod_{i=1}^{t} p(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|s_t,\boldsymbol{B}) = \underbrace{p(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|s_t,\boldsymbol{B})}_{f(s_t)} \underbrace{\prod_{i=1}^{t-1} p(\bar{\boldsymbol{y}}_{\boldsymbol{i}}|s_i,\boldsymbol{B})}_{g(s_t,\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}})}$$

We have that:

$$p(s_t = i,\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}},\bar{\boldsymbol{y}}_{\boldsymbol{T}}|\theta) = p(s_t = i,\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}},\bar{\boldsymbol{y}}_{\boldsymbol{1:t}}|\theta) = \overbrace{\underbrace{p(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|\, s_t = i,\boldsymbol{\theta})}_{p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|\,\bar{b}_i)}}^{f(s_t)} \overbrace{P(\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}},\bar{\boldsymbol{y}}_{\boldsymbol{1:t-1}},|\boldsymbol{\theta})}^{g(s_t,\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}})}$$

In order to implement the algorithm we define the value $\delta_t(i)$ as:

$$\begin{aligned}\delta_t(i) = \max_{\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}}}\{p(s_t = i,\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}},\bar{\boldsymbol{y}}_{\boldsymbol{1:t}}|\theta)\} &= \max_{\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}}}\left\{\overbrace{\underbrace{p(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|\, s_t = i,\boldsymbol{\theta})}_{p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|\,\bar{b}_i)}}^{f(s_t)} \overbrace{P(\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}},\bar{\boldsymbol{y}}_{\boldsymbol{1:t-1}},|\boldsymbol{\theta})}^{g(s_t,\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}})}\right\} \\ &= p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|\,\bar{\boldsymbol{b}}_{\boldsymbol{i}}) \max_{s_{t-1}}\left\{\max_{\bar{s}_{1:t-2}}\{P(\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}},\bar{\boldsymbol{y}}_{\boldsymbol{1:t-1}},|\boldsymbol{\theta})\}\right\} = p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|\,\bar{b}_i)\max_{s_{t-1}}\{\delta_{t-1}(j)\} \\ &= \boldsymbol{p_i(\bar{y}_t|\,\bar{b}_i)\max_j\{\delta_{t-1}(i)\}}\end{aligned}$$

And so we have the equations:

$$\begin{aligned}&\delta_1^n(i) = p_i(\bar{y}_1^n|\,\bar{b}_i) && i = 1,\dots,I \quad n = 1,\dots,N \\ &\delta_t^n(i) = p_i(\bar{y}_t^n|\,\bar{b}_i)\boldsymbol{\max_j\{\delta_{t-1}(j)\}} && i = 1,\dots,I \quad t = 2,\dots,T \quad n = 1,\dots,N\end{aligned}$$

We start building the survival paths from $t = 1$, to $t = T$.
At every step t, the survival path that ends in $s_t = i$ will be made of the joining between

- The state $s_t = i$.
- The Survival Path that ended in $s_{t-1} = j$ such that $j = \arg\max_j\{\delta_t(i)\}$

When we are done, we will end up with I survival paths, each one ending in a different $s_T = i$
We pick the best one of them as the best solution.

3.5.4 Viterbi's Algorithm for MAP decoder

The maximization problem we have is:

$$\max_{\boldsymbol{S}}\{p(\boldsymbol{S},\boldsymbol{Y}|\boldsymbol{\theta})\} = \max_{\boldsymbol{s_{1:T}}}\{p(\boldsymbol{s_{1:T}},\bar{\boldsymbol{y}}_{\boldsymbol{1:T}}|\boldsymbol{\theta})\}$$

If for any index t, we can express $p(\boldsymbol{s_{1:T}},\bar{\boldsymbol{y}}_{\boldsymbol{1:T}}|\boldsymbol{\theta})$ *as*:

$$\underbrace{p(\bar{\boldsymbol{y}}_{\boldsymbol{1:T}}|\boldsymbol{s_{1:t}},\boldsymbol{\theta})}_{m(\boldsymbol{s_{1:t}})} = f(s_t)g(s_t,\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}})$$

In MAP case, we have:

$$p(\bar{\boldsymbol{y}}_{\boldsymbol{1:T}},\bar{\boldsymbol{s}}_{\boldsymbol{1:t}},\boldsymbol{\theta}) = \underbrace{\left(p(s_1|\bar{\boldsymbol{\theta}})\prod_{i=2}^{t}p(s_t|s_{t-1},\boldsymbol{A})\right)}_{p(S|\bar{\boldsymbol{\theta}})}\cdot\underbrace{\left(\prod_{t=1}^{T}p(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|s_t,\boldsymbol{B})\right)}_{p(Y|S,\bar{\boldsymbol{\theta}})}$$

We have that:

$$\begin{aligned}p(s_t = i,\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}},\bar{\boldsymbol{y}}_{\boldsymbol{T}}|\theta) &= p(s_t = i,\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}},\bar{\boldsymbol{y}}_{\boldsymbol{1:t}}|\theta)\\ &= \overbrace{\underbrace{p(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|\,s_t = i)}_{p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|\,\bar{b}_i)}}^{f(s_t)}\overbrace{\underbrace{P(s_t = i|s_{t-1})}_{a_{ji}}P(\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}},\bar{\boldsymbol{y}}_{\boldsymbol{1:t-1}})}^{g(s_t,\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}})}\end{aligned}$$

In order to implement the algorithm we define the value $\delta_t(i)$ as

$$\begin{aligned}\delta_t(i) &= \max_{s_{1:t-1}}\{p(s_t = i,\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}},\bar{\boldsymbol{y}}_{\boldsymbol{1:t}}|\theta)\} \quad [\textit{From calculating } \alpha_t(i) \textit{ we know}]\\ &= \max_{s_{1:t-1}}\left\{\overbrace{\underbrace{p(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|\,s_t = i)}_{p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|\,\bar{b}_i)}}^{f(s_t)}\overbrace{\underbrace{P(s_t = i|s_{t-1})}_{a_{ji}}P(s_{t-1},\bar{\boldsymbol{y}}_{\boldsymbol{1:t-1}})}^{g(s_t,\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}})}\right\}\\ &= p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|\,\bar{b}_i)\max_{s_{t-1}}\left\{a_{ji}\max_{s_{1:t-2}}\{P(\bar{\boldsymbol{s}}_{\boldsymbol{1:t-1}},\bar{\boldsymbol{y}}_{\boldsymbol{1:t-1}})\}\right\}\\ &= p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|\,\bar{b}_i)\max_{s_{t-1}}\{a_{ji}\,\delta_{t-1}(i)\} = \boldsymbol{p_i(\bar{y}_t|\,\bar{b}_i)\max_j\{a_{ji}\,\delta_{t-1}(i)\}}\end{aligned}$$

And so we have the equations:

$$\begin{aligned}&\delta_1^n(i) = \pi_i\cdot p_i(\bar{y}_1^n|\,\bar{b}_i) && i = 1,\dots,I \quad n = 1,\dots,N\\ &\delta_t^n(i) = p_i(\bar{y}_t^n|\,\bar{b}_i)\boldsymbol{\max_j\{a_{ji}\,\delta_{t-1}(i)\}} && i = 1,\dots,I \quad t = 2,\dots,T \quad n = 1,\dots,N\end{aligned}$$

We start building the survival paths from $t = 1$, to $t = T$.
At every step t, the survival path that ends in $s_t = i$ will be made of the joining between

- The state $s_t = i$.
- The Survival Path that ended in $s_{t-1} = j$ such that $j = \arg\max_j\{\delta_t(i)\}$

When we are done, we will end up with I survival paths, each one ending in a different $s_T = i$
We pick the best one of them as the best solution.

3.6 Baum and Welch Algorithm

Baum-Welch is a specific case of the more general Expectation-Maximization (EM) algorithm.
It is an algorithm for learning model parameters $\boldsymbol{\theta} = \{\bar{\boldsymbol{\pi}}, \boldsymbol{A}, \boldsymbol{B}\}$ of a Hidden Markov Model (HMM) by means of a set of N realizations of the HMMs $\boldsymbol{Y} = \{\boldsymbol{Y^1}, \boldsymbol{Y^2}, \dots, \boldsymbol{Y^N}\}$, each realization has T time index elements, $\boldsymbol{Y^n} = \bar{\boldsymbol{y}}^{\boldsymbol{n}}_{\boldsymbol{1:T}} = \{\bar{\boldsymbol{y}}^{\boldsymbol{n}}_{\boldsymbol{1}}, \bar{\boldsymbol{y}}^{\boldsymbol{n}}_{\boldsymbol{2}}, \dots, \bar{\boldsymbol{y}}^{\boldsymbol{n}}_{\boldsymbol{T}}\}$.

The problem is not trivial since we don't have the state values $\boldsymbol{S^n} = \{\boldsymbol{s}^{\boldsymbol{n}}_{\boldsymbol{1}}, \boldsymbol{s}^{\boldsymbol{n}}_{\boldsymbol{2}}, \dots, \boldsymbol{s}^{\boldsymbol{n}}_{\boldsymbol{T}}\}$ associated to every observation. If we had the state $\boldsymbol{s_t}$ that generated every observation , $\bar{\boldsymbol{y}}_{\boldsymbol{t}}$ we would have the tupla $\{\boldsymbol{s_t}, \bar{\boldsymbol{y}}_{\boldsymbol{t}}\}$ and we could compute θ^* as the argument that maximizes the complete log-likelihood as:

$$\theta^* = arg\max_{\theta}\{\log(p(\boldsymbol{S}, \boldsymbol{Y}|\boldsymbol{\theta}))\} = arg\max_{\theta}\left\{\sum_{t=1}^{T}\log(p(s_t, \boldsymbol{Y}|\boldsymbol{\theta}))\right\}$$

This is not tractable, since there are DT^I different values of $\boldsymbol{S} = \{s_1, s_2, \dots, s_T\}$ to try.

Baum-Welch is an iterative procedure for estimating θ^* from only $\boldsymbol{Y} = \{\bar{\boldsymbol{y}}_{\boldsymbol{1}}, \bar{\boldsymbol{y}}_{\boldsymbol{2}}, \dots, \bar{\boldsymbol{y}}_{\boldsymbol{T}}\}$.
It works by maximizing a proxy to the log-likelihood, and updating the current model to be closer to the optimal model.
Each iteration of Baum-Welch is guaranteed to increase the complete log-likelihood of the data. but of course, convergence to the optimal solution is not guaranteed.

Baum-Welch can be described simply as repeating the following steps until convergence

- Expected complete data log likelihood (Expectation step or E step)

$$Q(\theta, \theta^{t-1}) = E\{l_c(\theta)|\mathcal{D}, \theta^{t-1}\}$$

- Maximization step (M step)
  - ML Estimation

$$\theta^t = \underset{\theta}{\text{argmax}} Q(\theta, \theta^{t-1})$$

  - MAP Estimation

$$\theta^t = \underset{\theta}{\text{argmax}} Q(\theta, \theta^{t-1}) + \ln p(\theta)$$

- Convergence: $l_c(\theta^t) \geq l_c(\theta^{t-1})$

3.6.1 Complete Log Likelihood of the data

For a given realization of the HMM $\boldsymbol{Y} = \{\bar{\boldsymbol{y}}_{\boldsymbol{1}}, \bar{\boldsymbol{y}}_{\boldsymbol{2}}, \dots, \bar{\boldsymbol{y}}_{\boldsymbol{T}}\}$ the complete log-likelihood of the model for that realization is:

$$I_C(\bar{\boldsymbol{\theta}}) = \ln\big(p(\boldsymbol{Y}, \boldsymbol{S}|\boldsymbol{\theta})\big)$$

We have by the Chain Rule that:

$$p(\boldsymbol{Y}, \boldsymbol{S}|\boldsymbol{\theta}) = p(\boldsymbol{S}|\boldsymbol{\theta}) p(\boldsymbol{Y}|\boldsymbol{S}, \boldsymbol{\theta})$$

Where

- $p(S|\bar{\boldsymbol{\theta}})$: Probability of generating the sequence of states $\boldsymbol{S} = \{\boldsymbol{s}_{\boldsymbol{1}}, \boldsymbol{s}_{\boldsymbol{2}}, \dots, \boldsymbol{s}_{\boldsymbol{T}}\}$

  By the Chain Rule:

$$\begin{aligned} p(S|\bar{\boldsymbol{\theta}}) &= p(s_1, s_2, \dots, s_T|\boldsymbol{\theta}) \\ &= p(s_1|\boldsymbol{\theta}) \cdot p(s_2|s_1, \boldsymbol{\theta}) \cdot p(s_3|s_2, s_1, \boldsymbol{\theta}) \cdots \; p(s_T|s_{T-1}, \dots, s_1, \boldsymbol{\theta}) \end{aligned}$$

  Applying the properties of the Markov chain:

$$p(S|\boldsymbol{\theta}) = p(s_1|\boldsymbol{\theta}) \prod_{t=2}^{T} p(s_t|s_{t-1}, \boldsymbol{\theta}) = p(s_t|\boldsymbol{\theta}) \prod_{t=2}^{T} p(s_t|s_{t-1}, \boldsymbol{A})$$

- $p(Y|S, \boldsymbol{\theta})$: Probability of observing $\boldsymbol{Y} = \{\bar{\boldsymbol{y}}_{\boldsymbol{1}}, \bar{\boldsymbol{y}}_{\boldsymbol{2}}, \dots, \bar{\boldsymbol{y}}_{\boldsymbol{T}}\}$ given the states $\boldsymbol{S}$
-

$$p(Y|S, \boldsymbol{\theta}) = p(\bar{\boldsymbol{y}}_{\boldsymbol{1}}, \bar{\boldsymbol{y}}_{\boldsymbol{2}}, \dots, \bar{\boldsymbol{y}}_{\boldsymbol{T}}|s_1, s_2, \dots, s_T, \boldsymbol{\theta}) = \prod_{t=1}^{T} p(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|s_t, \boldsymbol{\theta}) = \prod_{t=1}^{T} p(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|s_t, \boldsymbol{B})$$

So we have that, for a given sequence of observations $\boldsymbol{Y} = \{\bar{\boldsymbol{y}}_{\boldsymbol{1}}, \bar{\boldsymbol{y}}_{\boldsymbol{2}}, \dots, \bar{\boldsymbol{y}}_{\boldsymbol{T}}\}$:

$$p(Y, S|\bar{\boldsymbol{\theta}}) = \underbrace{\left(p(s_1|\bar{\boldsymbol{\theta}}) \prod_{t=2}^{T} p(s_t|s_{t-1}, \boldsymbol{A})\right)}_{p(S|\bar{\boldsymbol{\theta}})} \cdot \underbrace{\left(\prod_{t=1}^{T} p(\bar{\boldsymbol{y}}_{\boldsymbol{t}}|s_t, \boldsymbol{B})\right)}_{p(Y|S, \bar{\boldsymbol{\theta}})}$$

(The 2 products are independent of each other)

If we have N independent realizations of the HMM $\boldsymbol{Y^n} = \{\bar{\boldsymbol{y}}_{\boldsymbol{1}}^{\boldsymbol{n}}, \bar{\boldsymbol{y}}_{\boldsymbol{2}}^{\boldsymbol{n}}, \dots, \bar{\boldsymbol{y}}_{\boldsymbol{T}}^{\boldsymbol{n}}\}$, with $\boldsymbol{Y} = \{\boldsymbol{Y^1}, \boldsymbol{Y^2}, \dots, \boldsymbol{Y^N}\}$ the complete likelihood of getting that set of realizations is:

$$p(Y, S|\bar{\boldsymbol{\theta}}) = \prod_{n=1}^{N} p(Y^n, S^n|\bar{\boldsymbol{\theta}}) = \prod_{n=1}^{N} \left(\left(p(s_1^n|\bar{\boldsymbol{\theta}}) \prod_{t=2}^{T} p(s_t^n|s_{t-1}^n, \boldsymbol{A})\right)\left(\prod_{t=1}^{T} p(\bar{\boldsymbol{y}}_{\boldsymbol{t}} = \bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|s_t^n, \boldsymbol{B})\right)\right)$$

The complete loglikelihood of the N realizations is:

$$I_C(\boldsymbol{\theta}) = \ln\big(p(Y, S|\boldsymbol{\theta})\big) = \sum_{n=1}^{N} \ln\big(p(Y = Y^n, S|\boldsymbol{\theta})\big)$$

With:

$$\ln\big(p(Y = Y^n, S|\boldsymbol{\theta})\big) = \ln\big(p(s_1^n|\boldsymbol{\theta})\big) + \sum_{t=2}^{T} \ln\big(p(s_t^n|s_{t-1}^n, \boldsymbol{A})\big) + \sum_{t=1}^{T} \ln\big(p(\bar{\boldsymbol{y}}_{\boldsymbol{t}} = \bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|s_t^n, \boldsymbol{B})\big)$$

We use the same trick as in the EM, expressing the probabilities in exponential form with Indicator function $\mathbb{I}(condition)$:

$$p(s_1^n|\bar{\boldsymbol{\theta}}) = \prod_{i=1}^{I}\left(p(s_1^n = i|\boldsymbol{\theta})\right)^{\mathbb{I}(s_1^n=i)}$$

$$p(s_t^n|s_{t-1}^n, \boldsymbol{A}) = \prod_{i=1}^{I}\prod_{j=1}^{I}\left(p(s_t^n = j|s_{t-1}^n = i, \boldsymbol{A})\right)^{\mathbb{I}(s_t^n=j, s_{t-1}^n=i\ )} \quad [watch\ the\ order\ j,i]$$

$$p(\bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|s_t^n, \boldsymbol{B}) = \prod_{i=1}^{I} p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}} = \bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|s_t^n = i, \boldsymbol{B})^{\mathbb{I}(s_t^n=i)}$$

We end up with the expression:

$$\ln\left(p(Y = Y^n, S|\bar{\boldsymbol{\theta}})\right)$$
$$= \sum_{i=1}^{I}\mathbb{I}(s_1^n = i)\ln\left(p(s_1^n = i|\boldsymbol{\theta})\right)$$
$$+ \sum_{t=2}^{T}\left(\sum_{i=1}^{I}\sum_{j=1}^{I}\mathbb{I}(s_t^n = i, s_{t-1}^n = j\ )\ln\left(p(s_t^n = j|s_{t-1}^n = i, \boldsymbol{A})\right)\right)$$
$$+ \sum_{t=1}^{T}\left(\sum_{i=1}^{I}\mathbb{I}(s_t^n = i)\ln\left(p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}} = \bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|s_t^n = i, \boldsymbol{B})\right)\right)$$

And so, the complete log-likelihood of the N realizations is:

$$I_C(\bar{\boldsymbol{\theta}}) = \ln\left(p(Y, S|\bar{\boldsymbol{\theta}})\right) = \sum_{n=1}^{N}\ln\left(p(Y = Y^n, S|\boldsymbol{\theta})\right) =$$
$$= \sum_{n=1}^{N}\sum_{i=1}^{I}\mathbb{I}(s_1^n = i)\underbrace{\ln\left(p(s_1^n = i|\boldsymbol{\theta})\right)}_{\ln(\pi_i)}$$
$$+ \sum_{n=1}^{N}\sum_{t=2}^{T}\left(\sum_{i=1}^{I}\sum_{j=1}^{I}\mathbb{I}(s_t^n = j, s_{t-1}^n = i\ )\underbrace{\ln\left(p(s_t^n = j|s_{t-1}^n = i, \boldsymbol{A})\right)}_{\ln(a_{ij})}\right)$$
$$+ \sum_{n=1}^{N}\sum_{t=1}^{T}\left(\sum_{i=1}^{I}\mathbb{I}(s_t^n = i)\underbrace{\ln\left(p(\bar{\boldsymbol{y}}_{\boldsymbol{t}} = \bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|s_t^n = i, \boldsymbol{B})\right)}_{\ln\left(p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|\bar{b}_i)\right)}\right)$$

We can swap the position of the sums as we desire, so we put first the sums over the domain of the states, and expressing the probabilities in terms of the HMM notation:

$$I_C(\boldsymbol{\theta}) = \sum_{i=1}^{I}\left(\sum_{n=1}^{N}\mathbb{I}(s_1^n = i)\ln(\pi_i)\right) + \sum_{i=1}^{I}\sum_{j=1}^{I}\left(\sum_{n=1}^{N}\sum_{t=2}^{T}\mathbb{I}(s_t^n = j, s_{t-1}^n = i\ )\ln(a_{ij})\right)$$
$$+ \sum_{i=1}^{I}\left(\sum_{n=1}^{N}\sum_{t=1}^{T}\mathbb{I}(s_t^n = i)\ln\left(p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|\bar{b}_i)\right)\right)$$

3.6.2 Expected Complete log likelihood

In order to perform the Baum and Welch algorithm we need to compute the Expected Complete Log Likelihood as:
We use variable r to denote the $r-th$ iteration of the algorithm

$$Q(\boldsymbol{\theta},\boldsymbol{\theta^{r-1}}) = E\{I_C(\boldsymbol{\theta})|Y,\boldsymbol{\theta^{r-1}}\}$$
$$= \sum_{i=1}^{I}\left(\sum_{n=1}^{N} E\{\mathbb{I}(s_1^n = i|Y,\boldsymbol{\theta^{r-1}})\}\ln(\pi_i)\right)$$
$$+ \sum_{i=1}^{I}\sum_{j=1}^{I}\left(\sum_{n=1}^{N}\sum_{t=2}^{T} E\{\mathbb{I}(s_t^n = j, s_{t-1}^n = i|Y,\boldsymbol{\theta^{r-1}})\}\ln(a_{ij})\right)$$
$$+ \sum_{i=1}^{I}\left(\sum_{n=1}^{N}\sum_{t=1}^{T} E\{\mathbb{I}(s_t^n = i|Y,\boldsymbol{\theta^{r-1}})\}\ln\left(p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|\bar{b}_i)\right)\right)$$

We have that:

$$E\{\mathbb{I}(s_1^n = i|\boldsymbol{Y},\boldsymbol{\theta^{r-1}})\} = P(s_1^n = i|\boldsymbol{Y},\boldsymbol{\theta^{r-1}}) = \gamma_1^n(i)$$
$$E\{\mathbb{I}(s_t^n = i|\boldsymbol{Y},\bar{\boldsymbol{\theta}}^{\boldsymbol{r-1}})\} = P(s_t^n = i|\boldsymbol{Y},\boldsymbol{\theta^{r-1}}) = \gamma_t^n(i)$$
$$E\{\mathbb{I}(s_t^n = j, s_{t-1}^n = i|\boldsymbol{Y},\bar{\boldsymbol{\theta}}^{\boldsymbol{r-1}})\} = P(s_t^n = j, s_{t-1}^n = i|\boldsymbol{Y},\bar{\boldsymbol{\theta}}^{\boldsymbol{r-1}}) = \xi_{t-1}^n(i,j)$$

Where:

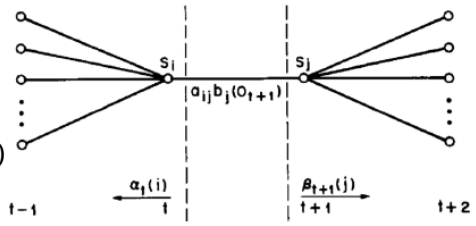
$$\gamma_t^n(i) \propto \alpha_t^n(i)\cdot\beta_t^n(i)$$
$$\xi_t^n(i,j) \propto \alpha_t^n(i)a_{ij}p_j(\bar{\boldsymbol{y}}_{\boldsymbol{t+1}}^{\boldsymbol{n}}|\bar{b}_j)\cdot\beta_{t+1}^n(j)$$

The constant that related those proportions can be found easily since, given a realization n and an instant t, we have:

$$\sum_{j=1}^{I}\gamma_t^n(i) = 1 \quad and \quad \sum_{i=1}^{I}\sum_{j=1}^{I}\xi_t^n(i,j) = 1$$

We define $\xi_t^n(i,j)$ as:
$\xi_t^n(i,j)$: Probability of being in state i at time t and being in state j at time $t+1$.

$$\xi_t^n(i,j) \propto \alpha_t^n(i)a_{ij}p_j(\bar{\boldsymbol{y}}_{\boldsymbol{t+1}}^{\boldsymbol{n}}|\bar{b}_j)\cdot\beta_{t+1}^n(j)$$
$$t = 1,..,T-1$$



We finally have the expression:

$$Q(\boldsymbol{\theta},\boldsymbol{\theta^{r-1}}) = E\{I_C(\boldsymbol{\theta})|Y,\boldsymbol{\theta^{r-1}}\}$$
$$= \sum_{i=1}^{I}\left(\sum_{n=1}^{N}\gamma_1^n(i)\ln(\pi_i)\right) + \sum_{i=1}^{I}\sum_{j=1}^{I}\left(\sum_{n=1}^{N}\sum_{t=2}^{T}\xi_{t-1}^n(i,j)\ln(a_{ij})\right)$$
$$+ \sum_{i=1}^{I}\left(\sum_{n=1}^{N}\sum_{t=1}^{T}\gamma_t^n(i)\ln\left(p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|\bar{b}_i)\right)\right)$$

3.6.1 Maximization Step

In the maximization step we will estimate the new model $\boldsymbol{\theta^r} = \{\overline{\boldsymbol{\pi}}, \boldsymbol{A}, \boldsymbol{B}\}^{\boldsymbol{r}}$ using the past parameters $\boldsymbol{\theta^{r-1}} = \{\overline{\boldsymbol{\pi}}, \boldsymbol{A}, \boldsymbol{B}\}^{\boldsymbol{r-1}}$ and the updated parameters $\{\gamma_t^n, \xi_t^n(i,j)\}$ values from the E-step.

The new parameters $\boldsymbol{\theta^r}$ are those that maximize $Q(\boldsymbol{\theta}, \boldsymbol{\theta^{r-1}})$ over all possible values of $\overline{\boldsymbol{\theta}}$

$$\boldsymbol{\theta^r} = \boldsymbol{arg}\ \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta^{r-1}})$$

As calculated before, the complete log-likelihood is expressed as:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta^{r-1}}) = \underbrace{\sum_{i=1}^{I}\left(\sum_{n=1}^{N}\gamma_1^n(i)\ln(\pi_i)\right)}_{Depends\ on\ \overline{\pi}} + \underbrace{\sum_{i=1}^{I}\sum_{j=1}^{I}\left(\sum_{n=1}^{N}\sum_{t=2}^{T}\xi_{t-1}^n(i,j)\ln(a_{ij})\right)}_{Depends\ on\ A}$$
$$+ \underbrace{\sum_{i=1}^{I}\left(\sum_{n=1}^{N}\sum_{t=1}^{T}\gamma_t^n(i)\ln(p_i(\overline{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|b_i))\right)}_{Depends\ on\ B}$$

So we need to maximize this over $\boldsymbol{\theta} = \{\overline{\boldsymbol{\pi}}_{\boldsymbol{K}}, \boldsymbol{A}, \boldsymbol{B}\}$.
Since $Q(\boldsymbol{\theta}, \boldsymbol{\theta^{r-1}})$ is a concave function over $\boldsymbol{\theta}$, we could perform the derivative with respect to $\boldsymbol{\theta}$ and set it to 0, to get the $\boldsymbol{\theta}$ that maximizes the function $Q(\boldsymbol{\theta}, \boldsymbol{\theta^{r-1}})$

$$\frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta^{r-1}})}{\partial \boldsymbol{\theta}}|_{\boldsymbol{\theta}=\boldsymbol{\theta^r}} = 0$$

3.6.1.1 Probabilities of the initial state $\bar{\pi}$

We are going to calculate the value of the initial probabilities $\bar{\boldsymbol{\pi}}$ that maximizes $Q(\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\theta}}^{\boldsymbol{r-1}})$ to obtain the next value of these coefficients $\bar{\boldsymbol{\pi}}^{\boldsymbol{r}}$:

$$\frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\boldsymbol{r-1}})}{\partial \bar{\boldsymbol{\pi}}}|_{\bar{\boldsymbol{\pi}}=\bar{\boldsymbol{\pi}}^{\boldsymbol{r}}} = 0$$

Since we are subject to the **constraint** that:

$$\sum_{i=1}^{I} \pi_i = 1$$

We add a **Lagrange multiplier** with that constraint so we must solve:

$$\frac{\partial}{\partial \bar{\boldsymbol{\pi}}}\left(\underbrace{\sum_{i=1}^{I}\left(\sum_{n=1}^{N} \gamma_1^n(i) \ln(\pi_i)\right)}_{Depends\ on\ \bar{\pi}_k} + \underbrace{\lambda\left(\sum_{i=1}^{I} \pi_i = 1\right)}_{Lagrange\ multiplier} \right) = 0$$

Taking partial derivatives with respect to the elements of $\bar{\boldsymbol{\pi}}$ we get, for the element i, π_i**:**

$$\frac{1}{\pi_i}\sum_{n=1}^{N} \gamma_1^n(i) + \lambda = 0$$

We have:

$$\pi_i = -\frac{\sum_{n=1}^{N} \gamma_1^n(i)}{\lambda} = -\frac{N_i}{\lambda} \qquad with\ \ N_i = \sum_{n=1}^{N} \gamma_1^n(i)$$

N_i : Expected number of times in state i at time $t = 1$

Since we have the constraint:

$$\sum_{i=1}^{I} \pi_i = 1 \ \rightarrow \qquad \sum_{i=1}^{I} -\frac{N_i}{\lambda} = 1 \ \rightarrow\ \lambda = -\sum_{i=1}^{I} N_i = -N \rightarrow\ N = \sum_{i=1}^{I}\sum_{n=1}^{N} \gamma_1^n(i)$$

So we finally have:

$$\pi_i = \frac{N_i}{N} = \frac{1}{N}\sum_{n=1}^{N} \gamma_1^n(i) \qquad N = \sum_{i=1}^{I}\sum_{n=1}^{N} \gamma_1^n(i)$$

3.6.1.1 Transition probabilities A

We are going to calculate the value of the initial probabilities $\bar{\boldsymbol{\pi}}$ that maximizes $Q(\boldsymbol{\theta}, \boldsymbol{\theta^{r-1}})$ to obtain the next value of these coefficients $\bar{\boldsymbol{\pi}}^{\boldsymbol{r}}$:

$$\frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta^{r-1}})}{\partial \boldsymbol{A}}|_{\boldsymbol{A}=\boldsymbol{A}^{\boldsymbol{r}}} = 0$$

Since we are subject to the **constraint** that the rows of A sum up to 1, because they are the probability of going from a state i, to any state j.

Let $\bar{a}_i = [a_{i1}, \dots, a_{ij}, \dots, a_{iI}]$ a row of A.

We add a **Lagrange multiplier** with that constraint so we must solve:

$$\frac{\partial}{\partial \bar{a}_i}\left(\underbrace{\sum_{i=1}^{I}\sum_{j=1}^{I}\left(\sum_{n=1}^{N}\sum_{t=2}^{T}\xi_t^n(i,j)\ln(a_{ij})\right)}_{Depends\ on\ A} + \underbrace{\lambda\left(\sum_{j=1}^{I}a_{ij}=1\right)}_{Lagrange\ multiplier}\right) = 0$$

Taking partial derivatives with respect to the elements of $\bar{a}_i$ we get, for the element j, a_{ij}**:**

$$\frac{1}{a_{ij}}\sum_{n=1}^{N}\sum_{t=2}^{T}\xi_t^n(i,j) + \lambda = 0$$

We have:

$$a_{ij} = -\frac{\sum_{n=1}^{N}\sum_{t=2}^{T}\xi_t^n(i,j)}{\lambda} = -\frac{E_{ij}}{\lambda} \quad with\ \ E_{ij} = \sum_{n=1}^{N}\sum_{t=2}^{T}\xi_{t-1}^n(i,j)$$

E_i : Expected number of transitions from state i to state j

Since we have the constraint:

$$\sum_{j=1}^{I}a_{ij} = 1 \ \rightarrow \ \sum_{j=1}^{I}-\frac{E_{ij}}{\lambda} = 1 \ \rightarrow \ \boldsymbol{\lambda = -\sum_{j=1}^{I}E_{ij} = -E_i} \ \rightarrow \ \boldsymbol{E_i = \sum_{j=1}^{I}\sum_{n=1}^{N}\sum_{t=2}^{T}\xi_{t-1}^n(i,j)}$$

So we finally have:

$$a_{ij} = \frac{E_{ij}}{E_i} = \frac{1}{E}\sum_{n=1}^{N}\sum_{t=2}^{T}\xi_{t-1}^n(i,j) \qquad E_i = \sum_{j=1}^{I}\sum_{n=1}^{N}\sum_{t=2}^{T}\xi_{t-1}^n(i,j)$$

3.6.1.1 Values of the parameters of the random distributions b

We are going to calculate the value of the parameter vector $\bar{\boldsymbol{b}}_{\boldsymbol{i}}$ of the i-th obsetantion probability function $p_i(\bar{\boldsymbol{y}}_y^n|\bar{\boldsymbol{b}}_{\boldsymbol{i}})$that maximizes $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\boldsymbol{r-1}})$ to obtain the next value of these coefficients $\bar{\boldsymbol{b}}_{\boldsymbol{i}}$:

$$\frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\boldsymbol{r-1}})}{\partial \bar{\boldsymbol{b}}_{\boldsymbol{i}}}|_{\bar{\boldsymbol{b}}_{\boldsymbol{i}}=\bar{\boldsymbol{b}}_{\boldsymbol{i}}^{\boldsymbol{r}}} = 0$$

Remember: $\boldsymbol{\theta} = \{\bar{\boldsymbol{\pi}}, \boldsymbol{A}, \boldsymbol{B}\}$ *and* $\bar{\bar{\boldsymbol{B}}}_{IxD} = [\bar{\boldsymbol{b}}_{\boldsymbol{1}}, \dots, \bar{\boldsymbol{b}}_{\boldsymbol{i}}, \dots, \bar{\boldsymbol{b}}_{\boldsymbol{I}}]$ so we are trying to get the value of one of there $\bar{\boldsymbol{b}}_{\boldsymbol{i}}$ vectors.

$$\frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\boldsymbol{r-1}})}{\partial \bar{\boldsymbol{b}}_{\boldsymbol{i}}}|_{\bar{\boldsymbol{b}}_{\boldsymbol{i}}=\bar{\boldsymbol{b}}_{\boldsymbol{i}}^{\boldsymbol{r}}} = \frac{\partial}{\partial \bar{\boldsymbol{b}}_{\boldsymbol{i}}}\left(\underbrace{\sum_{i=1}^{I}\left(\sum_{n=1}^{N}\sum_{t=1}^{T}\gamma_t^n(i)\ln\left(p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|\bar{\boldsymbol{b}}_{\boldsymbol{i}})\right)\right)}_{Depends\ on\ \bar{\boldsymbol{\theta}}_{\boldsymbol{k}}}\right) = 0$$

So, to get $\bar{\boldsymbol{\theta}}_{\boldsymbol{i}}$ we must solve the equation:

$$\sum_{n=1}^{N}\sum_{t=1}^{T}\gamma_t^n(i)\left[\frac{\partial}{\partial \bar{\boldsymbol{b}}_{\boldsymbol{k}}}\left(\ln\left(p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|\bar{\boldsymbol{b}}_{\boldsymbol{i}})\right)\right)\right] = 0$$

If $p_i(\bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|\bar{\boldsymbol{\theta}}_{\boldsymbol{i}})$ belongs to the exponential family, we have:

$$\frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\boldsymbol{r-1}})}{\partial \bar{\boldsymbol{b}}_{\boldsymbol{i}}}|_{\bar{\boldsymbol{b}}_{\boldsymbol{i}}=\bar{\boldsymbol{b}}_{\boldsymbol{i}}^{\boldsymbol{r}}} = \frac{\partial}{\partial \bar{\boldsymbol{b}}_{\boldsymbol{i}}}\left(\underbrace{\sum_{n=1}^{N}\sum_{t=1}^{T}\gamma_t^n(i)\left(\bar{\boldsymbol{b}}_{\boldsymbol{i}}^{\boldsymbol{T}}\cdot\phi_i(\bar{\boldsymbol{y}}_t^n) - A_i(\bar{\boldsymbol{b}}_{\boldsymbol{i}})\right)}_{Depends\ on\ \bar{\boldsymbol{b}}_{\boldsymbol{i}}}\right) = 0$$

(Same as in the exponential family but with the $\gamma_t^n(i)$ values)
As we have seen earlier, the moment matching of the exponential family implies:

$$\frac{\partial A(\bar{\boldsymbol{b}}_{\boldsymbol{i}})}{\partial \bar{\boldsymbol{b}}_{\boldsymbol{i}}} = E\{\phi(\bar{\boldsymbol{Y}}_i)\} = \frac{\boldsymbol{1}}{\boldsymbol{N}}\sum_{\boldsymbol{n=1}}^{\boldsymbol{N}}\boldsymbol{\phi_i}(\bar{\boldsymbol{y}}_t^n)$$

We have that:

$$\frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\boldsymbol{r-1}})}{\partial \bar{\boldsymbol{b}}_{\boldsymbol{i}}}|_{\bar{\boldsymbol{b}}_{\boldsymbol{i}}=\bar{\boldsymbol{b}}_{\boldsymbol{i}}^{\boldsymbol{r}}} = \sum_{n=1}^{N}\sum_{t=1}^{T}\gamma_t^n(i)(\phi_i(\bar{\boldsymbol{y}}_t^n) - E\{\phi_i(\bar{\boldsymbol{Y}}_i)\}) = 0$$

$$\sum_{n=1}^{N}\sum_{t=1}^{T}\gamma_t^n(i)\left(\phi(\bar{\boldsymbol{y}}_t^n)\right) = E\{\phi(\bar{\boldsymbol{Y}}_i)\}\sum_{n=1}^{N}\sum_{t=1}^{T}\gamma_t^n(i) \qquad Let\ \ \boldsymbol{\Gamma_i} = \sum_{n=1}^{N}\sum_{t=1}^{T}\gamma_t^n(i)$$

We arrive to the formula:

$$E\{\phi(\bar{\boldsymbol{Y}}_i)\} = \frac{1}{\boldsymbol{\Gamma_i}_i}\sum_{n=1}^{N}\sum_{t=1}^{T}\gamma_t^n(i)\left(\phi(\bar{\boldsymbol{y}}_t^n)\right)\ \ with\ \boldsymbol{\Gamma_i} = \sum_{n=1}^{N}\sum_{t=1}^{T}\gamma_t^n(i)$$

Moment Matching of the exponential family for the HMM.

3.7 Computation of the Baum and Welch

In this section we will see the pseudocode for computing the Baum and Welch Algorithm. First we will see the different Matrices of parameters needed to be calculated in the E step and then how to calculate the parameters of the model $\boldsymbol{\theta} = \{\bar{\boldsymbol{\pi}}, \boldsymbol{A}, \boldsymbol{B}\}$.

3.7.1 Expectation Step

As we have seen from the complete log-likelihood derivation, the parameters needed for calculating $Q(\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\theta}}^{\boldsymbol{t-1}}) = E\{I_C(\bar{\boldsymbol{\theta}})|D, \bar{\boldsymbol{\theta}}^{\boldsymbol{t-1}}\}$, using $\boldsymbol{\theta}^{\boldsymbol{r-1}} = \{\bar{\boldsymbol{\pi}}, \boldsymbol{A}, \boldsymbol{B}\}^{r-1}$ to do so are:

$\gamma_t^n(i)$: for $t = 1, \dots, T$; $n = 1, \dots, N$; $i = 1, \dots, I$
$\xi_{t-1}^n(i,j)$: for $t = 2, \dots, T$; $n = 1, \dots, N$; $i = 1, \dots, I$; $j = 1, \dots, I$

We have that these values are obtained as:

$$\gamma_t^n(i) = P(s_1^n = i|Y, \boldsymbol{\theta}^{\boldsymbol{r-1}}) \propto \alpha_t^n(i) \cdot \beta_t^n(i)$$
$$\xi_{t-1}^n(i,j) = P(s_t^n = i, s_{t-1}^n = j|Y, \bar{\boldsymbol{\theta}}^{\boldsymbol{r-1}}) \propto \alpha_{t-1}^n(i) \cdot a_{ij} p_j(\bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|\bar{b}_j) \cdot \beta_t^n(j)$$

With the values of alpha and beta obtained from the Forward-Backwards algorithm:

3.7.1.1 Obtaining the matrix of alphas, betas and gammas.

Formula for the alphas:

$$\alpha_1^n(i) = \pi_i \cdot p_i(\bar{y}_1^n|\bar{b}_i) \qquad i = 1, \dots, I \quad n = 1, \dots, N$$
$$\alpha_t^n(i) = p_i(\bar{y}_t^n|\bar{b}_i) \sum_{j=1}^{I} a_{ji} \alpha_{t-i}^n(j) \qquad i = 1, \dots, I \quad t = 2, \dots, T \quad n = 1, \dots, N$$

Formula for the betas:

$$\beta_T^n(i) = 1 \qquad i = 1, \dots, I \quad n = 1, \dots, N$$
$$\beta_t^n(i) = \sum_{j=1}^{I} a_{ij} \cdot p_j(\bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|\bar{\boldsymbol{b}}_{\boldsymbol{j}}) \cdot \beta_{t+1}^n(j) \qquad i = 1, \dots, I \quad t = 1, \dots, T-1 \quad n = 1, \dots, N$$

For a given realization of the HMM n, and a given time instant t:

$$\gamma_t^n(i) = P(s_t^n = i|\boldsymbol{Y}, \boldsymbol{\theta}) = \frac{\alpha_t^n(i) \cdot \beta_t^n(i)}{P(\boldsymbol{Y}|\boldsymbol{\theta})} \propto \alpha_t^n(i) \cdot \beta_t^n(i)$$

We don't care about $P(\boldsymbol{Y}|\boldsymbol{\theta})$ since we know that

$$\sum_{i=1}^{I} P(s_t^n = i|\boldsymbol{Y}, \boldsymbol{\theta}) = \sum_{i=1}^{I} \gamma_t^n(i) = 1$$

We could calculate $P(\boldsymbol{Y}|\boldsymbol{\theta})$, but it is easier to obtain it from this property.

- **Computing the matrix of alphas:**

We compute a matrix of alphas $\boldsymbol{\alpha(I,N,T)}$ where:

$$\alpha(i,n,t) = \alpha_t^n(i)$$

We compute $\alpha(i,:,:)$ from $i = 1$:N since $\alpha(i,n,t)$ needs $\alpha(:,n,t+1)$ to be obtained.

- **Computing the matrix of betas:**

We compute a matrix of betas $\boldsymbol{\beta(I,N,T)}$ where:

$$\beta(i,n,t) = \beta_t^n(i)$$

We compute $\alpha(i,:,:)$ from $i = N$:1 since $\alpha(i,n,t)$ needs $\alpha(:,n,t+1)$ to be obtained.

- **Computing the matrix of gammas:**

We compute a matrix of betas $\boldsymbol{\gamma(I,N,T)}$ where:

$$\gamma(i,n,t) = \gamma_t^n(i) = \alpha_t^n(i) \cdot \beta_t^n(i) = \alpha(i,n,t) \cdot \beta(i,n,t)$$

Never mind the order of computation.
After calculating this, normalize every $\gamma(:,n,t)$ vector, since:

$$\sum_{i=1}^{I} \gamma_t^n(i) = 1$$

3.7.1.1 Obtaining the matrix of fis ξ

We compute a matrix of fis $\boldsymbol{\xi(I,J,N,T-1)}$ where:

$$\xi(i,j,n,t) = \xi_t^n(i,j) = \alpha_t^n(i) \cdot a_{ij} p_j\left(\bar{\boldsymbol{y}}_{\boldsymbol{t+1}}^{\boldsymbol{n}} \middle| \bar{b}_j\right) \cdot \beta_{t+1}^n(j)$$

$$\xi(i,j,n,t) = \alpha(i,n,t) A(i,j) B(j, \bar{\boldsymbol{y}}_{\boldsymbol{t+1}}^{\boldsymbol{n}}) \beta(i,n,t+1)$$

Not defined for $t = T$
t =1 is not used either, since it is used for the initial probabilities.

After calculating this, normalize every $\xi(:,:,n,t)$ matrix, since:

$$\sum_{i=1}^{I} \sum_{j=1}^{I} \xi_t^n(i,j) = 1$$

3.7.1 Maximization Step

In the maximization step we will estimate the new model $\boldsymbol{\theta^r} = \{\bar{\boldsymbol{\pi}}, \boldsymbol{A}, \boldsymbol{B}\}^{\boldsymbol{r}}$ using the past parameters $\boldsymbol{\theta^{r-1}} = \{\bar{\boldsymbol{\pi}}, \boldsymbol{A}, \boldsymbol{B}\}^{\boldsymbol{r-1}}$ and the updated parameters $\{\gamma_t^n, \xi_t^n(i,j)\}$ values from the E-step.

The new parameters $\boldsymbol{\theta^r}$ are those that maximize $Q(\boldsymbol{\theta}, \boldsymbol{\theta^{r-1}})$ over all possible values of $\bar{\boldsymbol{\theta}}$

$$\boldsymbol{\theta^r} = \boldsymbol{arg}\ \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta^{r-1}})$$

As calculated before, the complete log-likelihood is expressed as:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta^{r-1}}) = \underbrace{\sum_{i=1}^{I}\left(\sum_{n=1}^{N} \gamma_1^n(i)\ln(\pi_i)\right)}_{Depends\ on\ \bar{\pi}} + \underbrace{\sum_{i=1}^{I}\sum_{j=1}^{I}\left(\sum_{n=1}^{N}\sum_{t=2}^{T} \xi_t^n(i,j)\ln(a_{ij})\right)}_{Depends\ on\ A} + \underbrace{\sum_{i=1}^{I}\left(\sum_{n=1}^{N}\sum_{t=1}^{T} \gamma_t^n(i)\ln(p_i(\overline{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}}|b_i))\right)}_{Depends\ on\ B}$$

We will now show the results we obtained earlier.

3.7.1.1 Obtaining the initial probabilities vector $\boldsymbol{\pi}$

The initial state probabilities $\bar{\boldsymbol{\pi}} = [\boldsymbol{\pi_1}, \boldsymbol{\pi_2}, \dots, \boldsymbol{\pi_I}]$ are obtained as:

$$\boldsymbol{\pi_i} = \frac{\boldsymbol{N_i}}{\boldsymbol{N}} = \frac{\boldsymbol{1}}{\boldsymbol{N}}\sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \boldsymbol{\gamma_1^n(i)}\ with \quad N = \sum_{i=1}^{I}\sum_{n=1}^{N} \gamma_1^n(i)$$

3.7.1.2 Obtaining the transition probabilities matrix $\boldsymbol{A}$

The transition probability matrix A is obtained as:

$$A = P = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1J} \\ a_{21} & a_{22} & \dots & a_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1} & a_{I2} & \dots & a_{IJ} \end{pmatrix} \quad a_{ij} = P(s_t = j | s_{t-1} = i) \qquad Rows\ Sum\ 1$$

$$a_{ij} = \frac{\boldsymbol{E_{ij}}}{\boldsymbol{E_i}} = \frac{\boldsymbol{1}}{\boldsymbol{E_i}}\sum_{n=1}^{N}\sum_{t=2}^{T} \xi_t^n(i,j) \quad with \quad \boldsymbol{E_i} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{I}}\sum_{n=1}^{N}\sum_{t=2}^{T} \xi_t^n(i,j)$$

3.7.1.3 Obtaining the parameters of the random distributions b

If $p_i(\bar{\boldsymbol{y}}|\bar{b}_i)$ belongs to the exponential family, we can obtain its parameters $\bar{\boldsymbol{b}}_{\boldsymbol{i}}$ from moment matching with the formula:

$$\boldsymbol{E}\{\boldsymbol{\phi}(\bar{\boldsymbol{Y}}_{\boldsymbol{i}})\} = \frac{\mathbf{1}}{\boldsymbol{\Gamma}_{\boldsymbol{i}_i}} \sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \sum_{\boldsymbol{t=1}}^{\boldsymbol{T}} \boldsymbol{\gamma}_{\boldsymbol{t}}^{\boldsymbol{n}}(\boldsymbol{i}) \left(\boldsymbol{\phi}(\bar{\boldsymbol{y}}_{\boldsymbol{t}}^{\boldsymbol{n}})\right) \ \ \boldsymbol{with}\ \boldsymbol{\Gamma}_{\boldsymbol{i}} = \sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \sum_{\boldsymbol{t=1}}^{\boldsymbol{T}} \boldsymbol{\gamma}_{\boldsymbol{t}}^{\boldsymbol{n}}(\boldsymbol{i})$$

$$\boldsymbol{B} = \begin{bmatrix} \bar{\boldsymbol{b}}_{\mathbf{1}} \\ \bar{\boldsymbol{b}}_{\mathbf{2}} \\ \vdots \\ \bar{\boldsymbol{b}}_{\boldsymbol{I}} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1D} \\ b_{21} & b_{22} & \cdots & b_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ b_{I1} & b_{I2} & \cdots & b_{ID} \end{bmatrix}$$

3.7.1.4 Obtaining the parameters of a multinomial distribution

Since we are given a family of D-dimensional observation functions:
Where $\bar{\boldsymbol{y}} = [y_1, y_2, .., y_D] \in \{0,1\}^D$

$$P_i(\bar{\boldsymbol{y}}|\bar{\boldsymbol{b}}_{\boldsymbol{i}}) = \prod_{d=1}^{D} b_{id}^{y_d}(1 - b_{id})^{1-y_d} = \begin{bmatrix} P(\bar{\boldsymbol{y}} = \bar{\boldsymbol{y}}_{\boldsymbol{i1}}) \\ P(\bar{\boldsymbol{y}} = \bar{\boldsymbol{y}}_{\boldsymbol{i2}}) \\ \vdots \\ P(\bar{\boldsymbol{y}} = \bar{\boldsymbol{y}}_{\boldsymbol{i}|\bar{\boldsymbol{y}}_{\boldsymbol{i}}|}) \end{bmatrix}$$

The vector of parameters of the i-th state is:

$$\bar{\boldsymbol{b}}_{\boldsymbol{i}} = [b_{i1}, b_{i2}, \ldots, b_{iD}]$$

$\bar{\boldsymbol{b}}_{\boldsymbol{i}}$ is the vector of averages of the independent D Bernoulli components of the multinomial.
We can obtain estimates for these parameters using the moment matching, and, as we have already seen:

$$E\{\phi(\bar{\boldsymbol{Y}}_i)\} = \frac{1}{\boldsymbol{\Gamma}_{\boldsymbol{i}_i}} \sum_{n=1}^{N} \sum_{t=1}^{T} \gamma_t^n(i) \left(\phi(\bar{\boldsymbol{y}}_t^n)\right) \ \ with\ \boldsymbol{\Gamma}_{\boldsymbol{i}} = \sum_{n=1}^{N} \sum_{t=1}^{T} \gamma_t^n(i)$$

So we have that:

$$b_{id} = \frac{1}{\boldsymbol{\Gamma}_{\boldsymbol{i}}} \sum_{n=1}^{N} \sum_{t=1}^{T} \gamma_t^n(i)(\boldsymbol{y}_{td}^n)$$