Assignment 2: HMM for Categorical Data Sequences

Daniel Barrejón Moreno

January 28, 2019

In this project we will develop a EM Baum-Welch algorithm for training HMM for the available data set, which are sequences of documents. The goal is to find the states or topics each document belongs to. In the following we will derive the equations and finally we will present the different results we obtained for different number of topics.

1 Complete data log-likehood $l_c(\boldsymbol{\theta})$ for the N sequences

The complete data log-likelihood has the following form

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

where log operator is the Naperian logarithm, S represents the hidden states of the model, Y is the observed continuous sequence, $\mathbf{A}$ stands for the state transition probabilities,B the observatoin emission probabilities and $\boldsymbol{\pi}$ is the initial state probability distribution.
The parameters of the model are

$$\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}, \quad (2)$$

where $\mathbf{A}$ is the state transition matrix, $\mathbf{B}$ is the emission matrix and $\boldsymbol{\pi}$ represents the probability of each state. In Equation 1 we can distinguish three main terms: the $\boldsymbol{\pi}$, $\mathbf{A}$ and $\mathbf{B}$ ones which can be rewritten as

$$p\left(s_1^n|\boldsymbol{\pi}\right) = \prod_{k=1}^{K} \pi_k^{\mathbb{I}\{s_1^n=k|Y,\boldsymbol{\theta}\}}, \quad (3)$$

$$p\left(s_t^n|s_{t-1}^n, \mathbf{A}\right) = \prod_{k=1}^{K} \prod_{k'=1}^{K} a_{k,k'}^{\mathbb{I}\{s_{t-1}^n=k, s_t^n=k'|Y,\boldsymbol{\theta}\}}, \quad (4)$$

$$p\left(\mathbf{y}_t^n|s_t^n, \mathbf{B}\right) = \prod_{k=1}^{K} p\left(\mathbf{y}_t^n|\boldsymbol{b}_k\right) = \prod_{k=1}^{K} p\left(\mathbf{y}_t^n|\boldsymbol{\theta}_k\right). \quad (5)$$

In this case, $\mathbb{I}$ represents an indicator function, $k = \{1, ..., K\}$ the current latent state of the model, $a_{k,k'}$ the kth row and k'th column element of the forementioned matrix $\mathbf{A}$ and $t = \{1, ..., T_n\}$ the position of the state in sequence n. In order to have a similar notation as the one used in Assignment 1, $\mathbf{b}_k$ (that belonged to $\mathbf{B}$) becomes $\boldsymbol{\theta}_k$, which represents the

categories of the topic k for the categorical distribution.

Since our data follow a categorical distribution, Equation 5 can be expressed as

$$p\left(\mathbf{y}_t^n|\boldsymbol{\theta}_k\right) = \prod_{j=1}^{Dt} \text{Cat}(y_{j,t}|\boldsymbol{\theta}_k), \tag{6}$$

and with further development we can get the following expression

$$p\left(\mathbf{y}_t^n|\boldsymbol{\theta}_k\right) = \prod_{j=1}^{Dt} \text{Cat}(y_{j,t}|\boldsymbol{\theta}_k) = \prod_{m=1}^{I}\prod_{j=1}^{Dt} \theta_{k,m}^{\mathbb{I}\{y_{j,t}^n=m\}} = \prod_{m=1}^{I} \theta_{k,m}^{\sum_{j=1}^{Dt}\mathbb{I}\{y_{j,t}^n=m\}} = \prod_{m=1}^{I} \theta_{k,m}^{\mu_{t,m}^n}, \tag{7}$$

where $\mu_{t,m}^n$ is denoted as follows

$$\mu_{t,m}^n = \sum_{j=1}^{Dt} \mathbb{I}\{y_{j,t}^n = m\}. \tag{8}$$

Finally, the expression of the complete data log-likelihood can be expressed in the following form

$$\begin{aligned} l_c\left(\boldsymbol{\theta}\right) = &\sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{I}\{s_1^n = k|Y,\boldsymbol{\theta}\}\log(\pi_k) + \\ &+ \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{k'=1}^{K}\sum_{t=1}^{T_n} \mathbb{I}\{s_{t-1}^n = k, s_t^n = k'|Y,\boldsymbol{\theta}\}\log(a_{k,k'}) + \\ &+ \sum_{n=1}^{N}\sum_{t=1}^{T_n} \mathbb{I}\{s_t^n = k|Y,\boldsymbol{\theta}\}\sum_{m=1}^{I}\mu_{t,m}^n\log(\theta_{k,m}). \end{aligned} \tag{9}$$

2 Expected Complete Data Logl-likelihood $Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}\right)$

The expected complete data log-likelihood has the following form

$$Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}\right) = E\left\{l_c(\boldsymbol{\theta})|\mathcal{D}, \boldsymbol{\theta}^{t-1}\right\} \tag{10}$$

As in the previous section, and considering the nature of l_c and its three main components, this expectation calculation can be divided in three.

$$\mathbb{E}\left(\sum_{n=1}^{N}\mathbb{I}\left(s_1^n = k|Y,\boldsymbol{\theta}\right)\right) = \sum_{n=1}^{N}\gamma_{n,1}(k), \tag{11}$$

$$\mathbb{E}\left(\sum_{n=1}^{N}\sum_{t=2}^{T_n}\mathbb{I}\left(s_{t-1}^n = k, s_t^n = k'|Y,\boldsymbol{\theta}\right)\right) = \sum_{n=1}^{N}\sum_{t=2}^{T_n}\xi_{n,t}(k,k'), \tag{12}$$

$$\mathbb{E}\left(\sum_{n=1}^{N}\sum_{t=1}^{T_n}\mathbb{I}\left(s_t^n = k|Y,\boldsymbol{\theta}\right)\right) = \sum_{n=1}^{N}\sum_{t=1}^{T_n}\gamma_{n,t}(k), \tag{13}$$

where it must be satisfied

$$\sum_{k=1}^{K} \gamma_{n,t}(k) = 1. \tag{14}$$

The analogous backwards term for$\gamma_{n,t}(k)$ is denoted as $\xi_{n,t}(k, k')$

$$\xi_{n,t}(k, k') = \alpha_{t-1}^{n}(k) a_{k,k'} \prod_{m=1}^{I} \theta_{k,m}^{\mu_{t,m}^{n}} \beta_{t}^{n}(k'), \tag{15}$$

and $\gamma_{n,t}(k)$

$$\gamma_{n,t}(k) \propto \beta_{t}^{n}(k) \alpha_{t}^{n}(k) \tag{16}$$

The terms α and β are computed by means of the **forward-backward algorithm** as follows

$$\alpha_{1}^{n}(k) = \pi_k \prod_{m=1}^{I} \theta_{k,m}^{\mu_{1,m}^{n}}, \tag{17}$$

$$\alpha_{t}^{n}(k) = \left(\sum_{k'=1}^{K} \alpha_{t-1}^{n}(k') a_{k',k} \right) \prod_{m=1}^{I} \theta_{k,m}^{\mu_{t,m}^{n}}, \tag{18}$$

$$\beta_{T_n}^{n}(k) = 1, \tag{19}$$

$$\beta_{t}^{n}(k) = \sum_{k'=1}^{K} a_{k,k'} \prod_{m=1}^{I} \theta_{k',m}^{\mu_{t+1,m}^{n}} \beta_{t+1}^{n}(k'). \tag{20}$$

The development above yields the final expression for $Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}\right)$

$$\begin{aligned} Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}\right) = & \sum_{k=1}^{K} \sum_{n=1}^{N} \gamma_{n,1}(k) \log(\pi_k) + \\ & + \sum_{k=1}^{K} \sum_{k'=1}^{K} \sum_{n=1}^{N} \sum_{t=2}^{T_n} \xi_{n,t}(k, k') \log(a_{k,k'}) + \\ & + \sum_{k=1}^{K} \sum_{m=1}^{I} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \gamma_{n,t}(k) \mu_{t,m}^{n} \log(\theta_{k,m}). \end{aligned} \tag{21}$$

3 ML Inference

In this section we develop the update formulas for the model, which are $\mathbf{A}$, $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$. These will be computed using Lagrange-Multipliers.

3.1 ML estimation of π_k

In order to compute π_k, we first need to take into account the following restrictions

$$0 \leq \pi_k \leq 1 \tag{22}$$

$$\sum_{k=1}^{K} \pi_k = 1. \tag{23}$$

Now, let us define the Lagrangian as

$$L\left(Q(\pi_k), \lambda\right) = Q(\pi_k) + \lambda\left(\sum_{k=1}^{K} \pi_k - 1\right), \tag{24}$$

which will be optimized in this way

$$\min_{\lambda} \max_{\pi_k} \{L\left(Q(\pi_k), \lambda\right)\}. \tag{25}$$

We first take the derivative with respect to π_k and equating it to 0

$$\begin{aligned} \frac{\partial L}{\partial \pi_k} &= \sum_{n=1}^{N} \frac{\gamma_{n,1}(k)}{\pi_k} - \lambda = 0, \\ \pi_k &= \frac{1}{\lambda} \sum_{n=1}^{N} \gamma_{n,1}(k). \end{aligned} \tag{26}$$

Now, with respect to λ

$$\begin{aligned} \frac{\partial L}{\partial \lambda} &= \sum_{k=1}^{K} \pi_k - 1 = 0, \\ \sum_{k=1}^{K} \pi_k &= 1. \end{aligned} \tag{27}$$

Taking into account the previous equation, if in both sides of Equation 26 summatories all over K are taken, then the value of λ can be obtained

$$\begin{aligned} \sum_{k=1}^{K} \pi_k &= \sum_{k=1}^{K} \frac{1}{\lambda} \sum_{n=1}^{N} \gamma_{n,1}(k) \\ 1 &= \frac{1}{\lambda} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{n,1}(k). \end{aligned} \tag{28}$$

Now, considering the previously imposed restrictions and recalling that

$$\sum_{k=1}^{K} \gamma_{n,t}(k) = 1, \tag{29}$$

the value of λ is

$$\begin{aligned} 1 &= \frac{1}{\lambda} \sum_{n=1}^{N} 1. \\ \lambda &= N. \end{aligned} \tag{30}$$

And with that value of λ the estimated value of π_k is

$$\widehat{\pi}_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_{n,1}(k). \tag{31}$$

3.2 ML estimation of $\theta_{k,m}$

Now, the parameter to be estimated is $\theta_{k,m}$ and the constraint is now

$$\sum_{m=1}^{I} \theta_{k,m} = 1. \tag{32}$$

In this case, the expression for the Lagrangian has the following form

$$L\left(Q(\theta_{k,m}), \lambda\right) = Q(\theta_{k,m}) + \lambda \left(\sum_{m=1}^{I} \theta_{k,m} - 1 \right), \tag{33}$$

which will be optimized in this way

$$\min_{\lambda} \max_{\theta_{k,m}} \{L\left(Q(\theta_{k,m}), \lambda\right)\}. \tag{34}$$

By first taking the derivative with respect to $\theta_{k,m}$ and equating it to 0

$$\begin{aligned} \frac{\partial L}{\partial \theta_{k,m}} &= \sum_{n=1}^{N} \sum_{t=1}^{T_n} \frac{\gamma_{n,t}(k)\mu_{t,m}^{n}}{\theta_{k,m}} - \lambda = 0, \\ \theta_{k,m} &= \frac{1}{\lambda} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \gamma_{n,t}(k)\mu_{t,m}^{n}. \end{aligned} \tag{35}$$

And later with respect to λ

$$\begin{aligned} \frac{\partial L}{\partial \lambda} &= \sum_{m=1}^{I} \theta_{k,m} - 1 = 0, \\ \sum_{m=1}^{I} \theta_{k,m} &= 1. \end{aligned} \tag{36}$$

Taking into account the previous equation, if in both sides of Equations 35 summatories all over I are taken, then the value of λ can be obtained

$$\begin{aligned} \sum_{m=1}^{I} \theta_{k,m} &= \sum_{m=1}^{I} \frac{1}{\lambda} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \gamma_{n,t}(k)\mu_{t,m}^{n} \\ 1 &= \frac{1}{\lambda} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \sum_{m=1}^{I} \gamma_{n,t}(k)\mu_{t,m}^{n}, \\ \lambda &= \sum_{n=1}^{N} \sum_{t=1}^{T_n} \sum_{m=1}^{I} \gamma_{n,t}(k)\mu_{t,m}^{n}. \end{aligned} \tag{37}$$

And with that value of λ the estimated value of $\theta_{k,m}$ is

$$\widehat{\theta}_{k,m} = \frac{\sum_{n=1}^{N} \sum_{t=1}^{T_n} \gamma_{n,t}(k)\mu_{t,m}^{n}}{\sum_{n=1}^{N} \sum_{t=1}^{T_n} \sum_{m=1}^{I} \gamma_{n,t}(k)\mu_{t,m}^{n}}. \tag{38}$$

3.3 ML estimation of $a_{k,k'}$

The last parameter to be estimated is $a_{k,k'}$. The constraint is now

$$\sum_{k'=1}^{K} a_{k,k'} = 1. \tag{39}$$

The Lagrangian will follow this expression

$$L\left(Q(a_{k,k'}), \lambda\right) = Q(a_{k,k'}) + \lambda \left(\sum_{k'=1}^{K} a_{k,k'} - 1 \right), \tag{40}$$

which will be optimized in this way

$$\min_{\lambda} \max_{a_{k,k'}} \{L\left(Q(a_{k,k'}), \lambda\right)\}. \tag{41}$$

By first taking the derivative with respect to $a_{k,k'}$ and equating it to 0

$$\begin{aligned} \frac{\partial L}{\partial a_{k,k'}} &= \sum_{n=1}^{N} \sum_{t=2}^{T_n} \frac{\xi_{n,t}(kk')}{a_{k,k'}} - \lambda = 0, \\ a_{k,k'} &= \frac{1}{\lambda} \sum_{n=1}^{N} \sum_{t=2}^{T_n} \xi_{n,t}(kk'). \end{aligned} \tag{42}$$

And later with respect to λ

$$\begin{aligned} \frac{\partial L}{\partial \lambda} &= \sum_{m=1}^{I} a_{k,k'} - 1 = 0, \\ \sum_{k'=1}^{K} a_{k,k'} &= 1. \end{aligned} \tag{43}$$

Taking into account the previous equation, if in both sides of (42) summatories all over I are taken, then the value of λ can be obtained

$$\begin{aligned} \sum_{k'=1}^{K} a_{k,k'} &= \sum_{k'=1}^{K} \frac{1}{\lambda} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \xi_{n,t}(k,k') \\ 1 &= \frac{1}{\lambda} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \sum_{k'=1}^{K} \xi_{n,t}(k,k'), \\ \lambda &= \sum_{n=1}^{N} \sum_{t=1}^{T_n} \sum_{k'=1}^{K} \xi_{n,t}(k,k'). \end{aligned} \tag{44}$$

And with that value of λ the estimated value of $a_{k,k'}$ is

$$\widehat{a}_{k,k'} = \frac{\sum_{n=1}^{N} \sum_{t=1}^{T_n} \xi_{n,t}(k,k')}{\sum_{n=1}^{N} \sum_{t=1}^{T_n} \sum_{k'=1}^{K} \xi_{n,t}(k,k')}. \tag{45}$$

4 State Decoder

Each document in each sequence has been generated by some state of the HMM. In the following section we present the two types of decoding algorithms that have been used to find such states: the state-by-state MAP decoding based on the Forward-Backwards algorithm and the ML iterative Viterbi algorithm.

4.1 MAP decoding based on Forward-Backwards

In order to determine the state of each document for the different sequences we obtain the most probable state for each document at each sequence from the parameter $\gamma_{n,t}(k)$ which is computed in the E-step of the Baum-Welch EM using Equation 16.

4.2 ML Viterbi decoding

As a comparison for the MAP decoding comment above, we have also implemented a Viterbi decoder. The Viterbi decoder is a dynamic programming algorithm for finding the most likely sequence of hidden states result of a sequence of observed events. Since we are dealing with hidden and observed states with HMM, the algorithm suits perfectly to our problem. The Viterbi decoding algorithm looks as follows

$$\underset{S}{\text{argmax}} p(S, Y) = \underset{i}{\text{argmax}} \left\{ \underset{s_{1:T-1}}{\text{argmax}} p\left(s_T = i, s_{1:T-1}, Y\right) \right\}. \tag{46}$$

In particular, we will use an iterative implementation of the Viterbi Algorithm. For such implementation we define two iterative steps described below.

- **Forward step**: Computed with the following equations

$$\delta_1(k) = \pi_k \prod_{m=1}^{I} \theta_{k,m}^{\mu_{1,m}^n} \quad 1 \leq k \leq K \tag{47}$$

$$\delta_t(k) = \prod_{m=1}^{I} \theta_{k,m}^{\mu_{t,m}^n} \max_{k'} a_{k',k} \delta_{t-1}(k') \quad 1 \leq k \leq K, 1 \leq t \leq T \tag{48}$$

$$\varphi_t(k) = \underset{k'}{\text{argmax}}\, a_{k',k} \delta_{t-1}(k') \quad 1 \leq k \leq K, 1 \leq t \leq T \tag{49}$$

$$\tag{50}$$

- **Backwards step**: The state estimation is computed using the following Equations

$$\hat{\mathbf{s}}_T = \arg\max_{\boldsymbol{k}} \delta_T(k) \tag{51}$$

$$\hat{s}_t = \varphi_{t+1}\left(\hat{s}_{t+1}\right) 1 \leq t \leq T. \tag{52}$$

5 Experiments

For the experiments conducted in this project the maximum number of iterations of the EM algorithm has been set to 100 iterations, the tolerance value has been set to 10^{-3} a total of 30 different initializations have been done. We have tested the system for different values of K, *i.e.* $K = \{2, 3, 4, 5\}$. For each of these experiments we will show the the best log-likelihood for the optimal initialization of the algorithm as well as A and π. The matrix $\boldsymbol{\theta}$ that is actually the model parameter $\mathbf{B}$ has not been shown because the number of rows is equal to the length of the dictionary I and displaying such matrix both in matrix form or with a colormap plot does not give much information of the emission probabilities.

One **important** aspect about the implementation of the algorithm is that instead of working with the actual $\xi_{n,t}(k, k')$ we work with a similar implementation used in [1], which consider working with the expected sufficient statistics for the transition matrix, for a given observation sequence. This may be rewritten as follows

$$\xi_{\Sigma}(k, k') = \sum_{t=2}^{T} p(S(t) = k, S(t+1) = k' | y(1:T)), \tag{53}$$

where the subscript Σ indicates the sum over t. Notice that, for a given sequence, this matrix is no longer a tensor but a matrix of dimension $K \times K$.

5.1 Simulations

In Table 1 we can check the different reported values of $\mathbf{A}$ and π for the different values of k.

$K =$	2	3	4	5
$\mathbf{A}$	$\begin{bmatrix} 0.32 & 0.68 \\ 0.37 & 0.63 \end{bmatrix}$	$\begin{bmatrix} 0.64 & 0.18 & 0.18 \\ 0.44 & 0.20 & 0.36 \\ 0.38 & 0.44 & 0.18 \end{bmatrix}$	$\begin{bmatrix} 0.2 & 0.27 & 0.43 & 0.1 \\ 0.21 & 0.30 & 0.13 & 0.36 \\ 0.1 & 0.35 & 0.34 & 0.21 \end{bmatrix}$	$\begin{bmatrix} 0.02 & 0.22 & 0.36 & 0.34 & 0.06 \\ 0.07 & 0.16 & 0.21 & 0.21 & 0.35 \\ 0.12 & 0.11 & 0.44 & 0.15 & 0.18 \\ 0.32 & 0.22 & 0.35 & 0.1 & 0.01 \\ 0.26 & 0.14 & 0.30 & 0.20 & 0.10 \end{bmatrix}$
π	$\begin{bmatrix} 0.28 & 0.72 \end{bmatrix}$	$\begin{bmatrix} 0.14 & 0.30 & 0.56 \end{bmatrix}$	$\begin{bmatrix} 0.34 & 0.38 & 0.28 & 1.27e^{-80} \end{bmatrix}$	$\begin{bmatrix} 0.48 & 0.38 & 6.46e^{-80} & 1.12e^{-103} & 0.13 \end{bmatrix}$

Table 1. Table for the different Experiments.

The comparison of the log-likelihood for each value of K is shown in Figure 1. As it can be seen, for greater values of K the log-likelihood improves. From this behaviour we could infer that the optimal value is $K = 5$. However, when we check the π values for $K = 4$ and $K = 5$ at Table 1 we can see that there is one and two topics respectively whose influence on the documents is actually a really small value which can be approximated to 0. Therefore, we can conclude that, although for $K = 5$ we get the optimal likelihood, actually when we observe the output result we could also conclude that $K = 3$ might be also a good candidate.

In Figure 2 we show individually each likelihood. As you might notice the likelihood always increases; however, when the curve gets to the 'plateau' behaviour, the likelihood tends

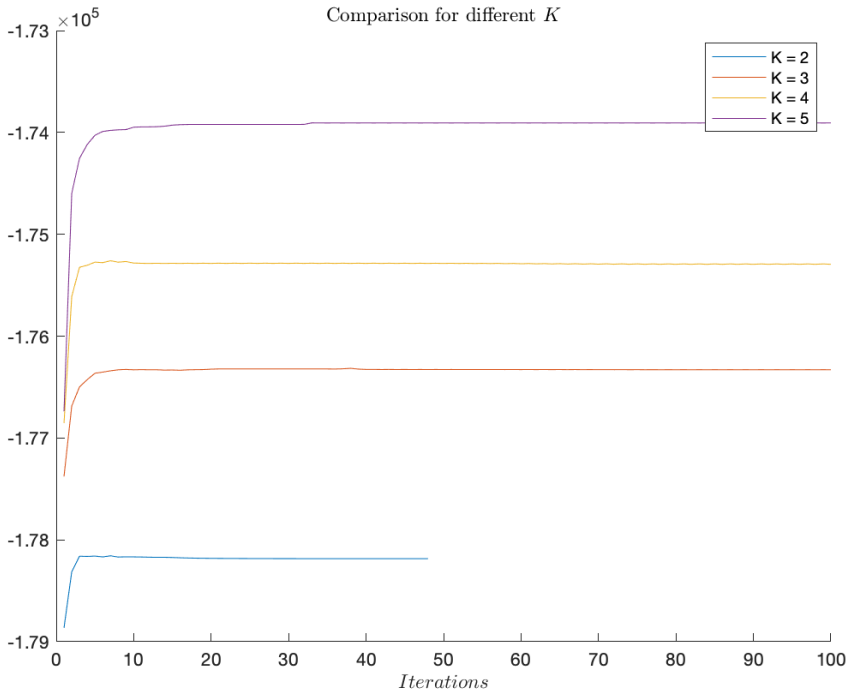


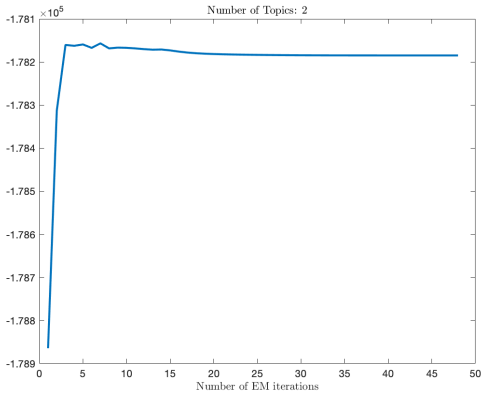
Figure 1. Comparison of the log-likelihoods for K $\in$ [2,5]

to oscillate. This is due to the correction steps we are taking into account in the implementation of the code when trying to keep the parameters of the model which are probabilities within the range $\in (0, 1)$. With some more depuration of the code this behaviour would disappear.

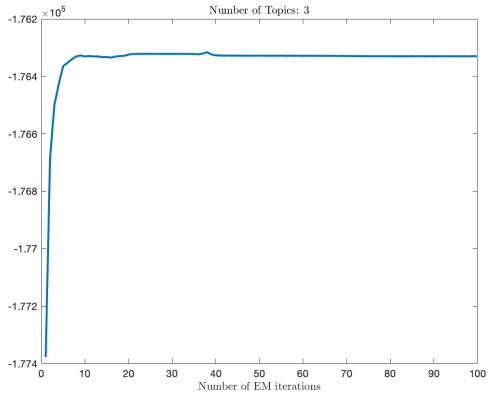
Finally, the states for each sequence and each document using both the MAP Forward-Backwards and the Viterbi algorithm are reported in the Figures 3, 4, 5 and 6.

6 Conclusions

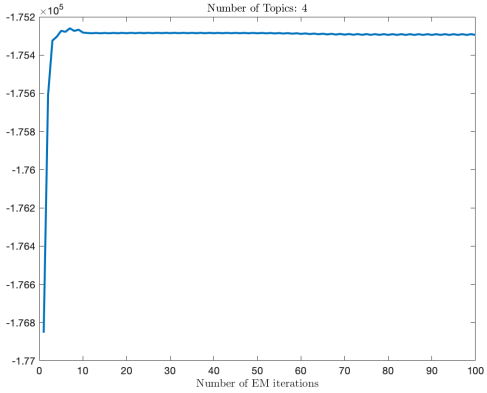
In this project we have implemented the EM Baum-Welch algorithm to apply it on a sequence of documents in order to find the topics they belong to. This is done using a HMM approach, for which we have been able to see the different states using the MAP state-by-state Forward-Backwards decoder and the Viterbi algorithm. Although our final reported likelihoods had some oscillations, still we could check that the likelihood improves at every iterations, and the final output with the model parameters is consistent to the solution we were looking for. However, we have not been able to compare the results of the decoding to the ground truth solution and therefore we cannot be certain whether the optimal number of topics is $K = 3$ or $K = 5$. What is clear is that every document is generated by a certain state and therefore it belongs to a very specific topic.



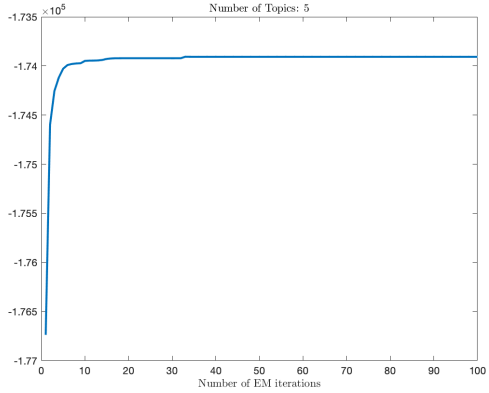
(a) Log-likelihood curve for K = 2



(b) Log-likelihood curve for K = 3



(c) Log-likelihood curve for K = 4



(d) Log-likelihood curve for K = 5

Figure 2. Separated log-likelihood curves for K ∈ [2,5]

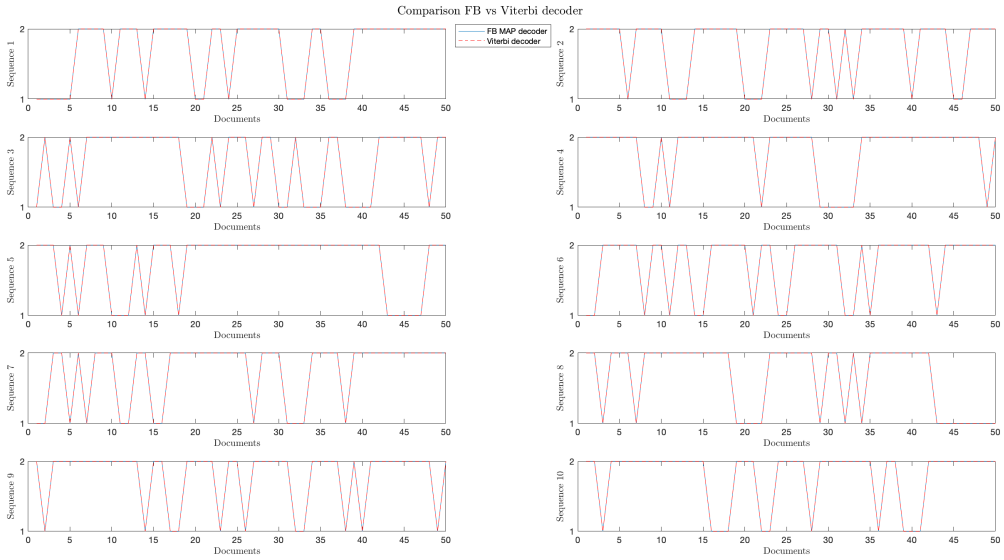


Figure 3. Forward Backward (MAP) and Viterbi (ML) estimations for K = 2

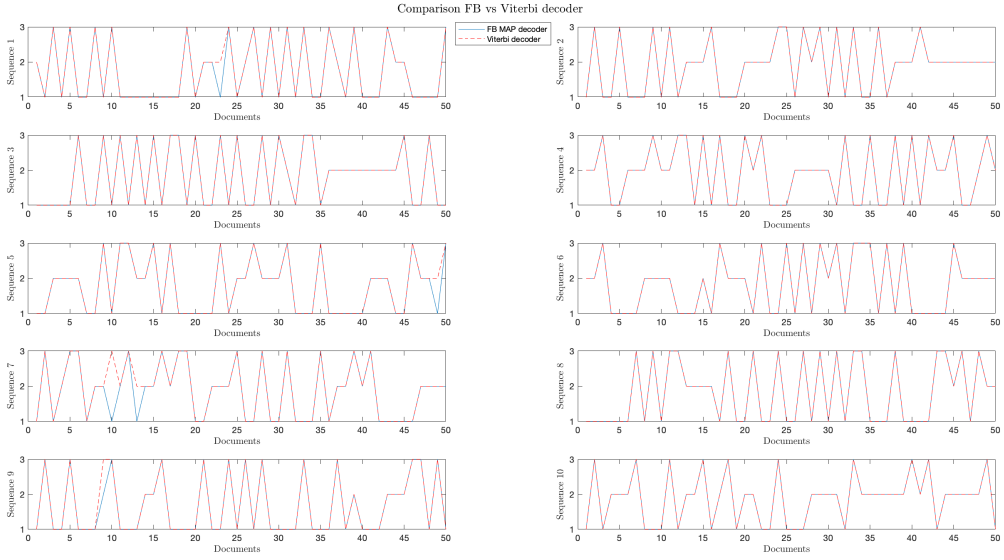


Figure 4. Forward Backward (MAP) and Viterbi (ML) estimations for K = 3

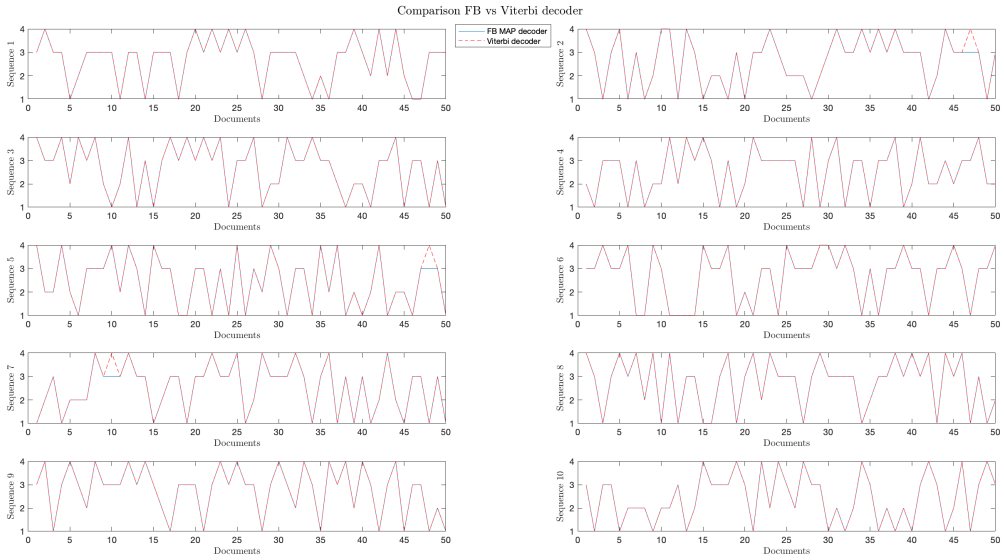


Figure 5. Forward Backward (MAP) and Viterbi (ML) estimations for K = 4

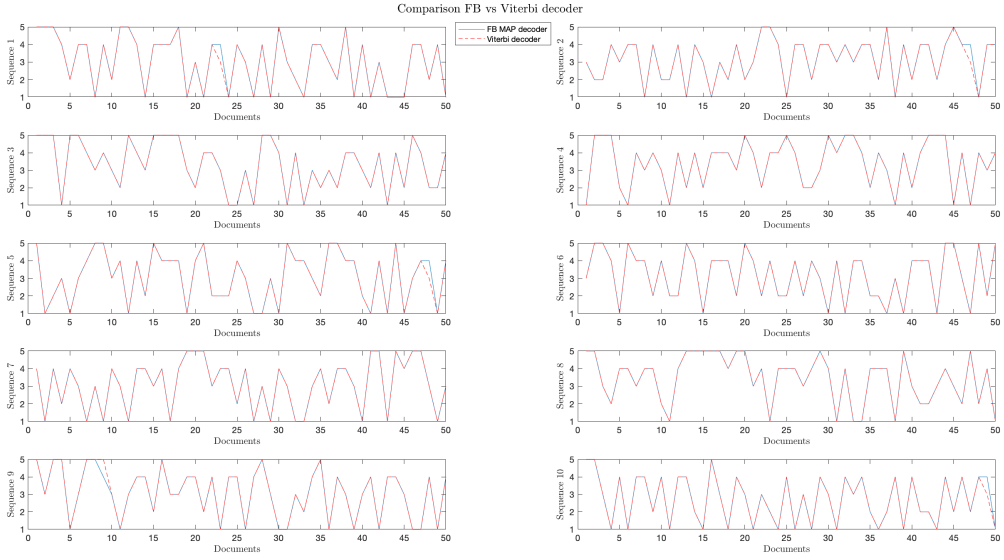


Figure 6. Forward Backward (MAP) and Viterbi (ML) estimations for K = 5

References

[1] K. P. Murphy, *Machine learning: A probabilistic perspective. adaptive computation and machine learning*, 2012.

[2] C. Fraley and A. E. Raftery, "Bayesian regularization for normal mixture estimation and model-based clustering," *Journal of classification*, vol. 24, no. 2, pp. 155–181, 2007.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006. [Online]. Available: http://research.microsoft.com/en-us/um/people/cmbishop/prml/.

[4] A. Artés-Rodríguez, *Notes for advanced signal processing*, 2018.

[5] R. Eisele, *The log-sum-exp trick in machine learning*. [Online]. Available: https://www.xarg.org/2016/06/the-log-sum-exp-trick-in-machine-learning/.