Clustering of Longitudinal Data via Probabilistic Models: the case of Semi-Markov Mixtures

Rosalie Millner

2024

Supervised by Madalina Olteanu

Dauphine UNIVERSITÉ PARIS | PSL CEREMADE UMR CNRS 7534

Contents

1 Introduction

In the last decades, sequence analysis has become a popular approach in sociology as a valuable tool for studying life trajectories. It allows researchers to analyse different types of longitudinal data, such as career trajectories, family dynamics, or life events, which are represented by sequences of categorical states over time. By studying these sequences, sociologists can extract useful insights out of seemingly complex data, by identifying patterns, trends, and differences across groups within a population, for example.

Nowadays, the traditional approach for sequence analysis is to rely on distance-based methods like Optimal Matching (OM). These methods involve selecting a dissimilarity-measure, which quantifies how different two sequences are, and applying it to all pairs of sequences within a dataset. Then, typically, hierarchical clustering is performed based on these distances, in order to group individuals into sub-populations. While such methods are widely used, they have several significant weaknesses that can cause problems in practice and impact the quality of results, notably their reliance on arbitrarily defined costs and their lack of sociological meaning.

An alternative approach is to explore probabilistic models. In recent years, methods based on Markov chains have emerged as a more robust and rigorous approach. This work presents a model based on finite mixtures of semi-Markov chains. The goal is to analyse longitudinal data described by categorical variables, and to cluster sequences into groups with similar trajectories. In contrast to traditional approaches that consider the entire sequence as a single unit of analysis, our semi-Markov model takes a step-by-step approach, estimating transition probabilities at each stage of a trajectory. This completely different approach allows greater understanding of the processes underlying sociological sequences. By using a mixture model, we can also account for the inherent heterogeneity within the population, enabling us to identify latent clusters of individuals.

I have studied and based my research on the paper "Estimating Finite Mixtures of Semi-Markov Chains: An Application to the Segmentation of Temporal Sensory Data" by Cardot et al. (2018) [1]. I have explored their model and implemented it in R[1]. In their study, the authors applied their mixture of semi-Markov chains to sensory data in the field of food science, which also involves sequences of categorical states over time. Here, I use it for the study of sociological life trajectories.

Drawing from this model, the various parameters are estimated using the Expectation-Maximisation (EM) algorithm. To determine the optimal number of clusters, the Bayesian Information Criterion (BIC) is used, which helps in selecting the appropriate number of mixture components, a number that is usually unknown in practice. Then, individuals are assigned to their respective clusters using the Maximum A Posteriori (MAP) classification criterion, ensuring that each sequence is grouped according to its highest probability of belonging to a specific cluster.

Compared to the traditional dissimilarity-based approaches, this probabilistic method offers several advantages. It not only provides a more robust model for the transitions between states, doing so without requiring one to define arbitrary costs, but also produces interpretable parameters that describe the behaviours within each cluster. This makes it possible to explain what characterizes the different groups and to understand the factors that distinguish one cluster from another, information that is valuable for a sociologist's study. In contrast, dissimilarity-based methods provide limited insight into the underlying dynamics of the sequences and often results in purely heuristic groupings.

[1] The code is available here: https://github.com/rosaliemillner/Clustering-longitudinal-data.

This paper is organized as follows: Section 2 presents an overview of the traditional approach used in sequence analysis, particularly focusing on Optimal Matching and its limitations. Section 3 introduces the theoretical framework of the finite mixture of semi-Markov chains as well as how we model the sequences. In Section 4, we explain how the various model parameters are estimated, as we detail the different steps of the algorithm. Finally, in Section 5, we perform a simulation study to evaluate the accuracy and performance of the model under various configurations, and were able to show that the results were promising.

2 Traditional Approach

A traditional application of sequence analysis involves 3 steps: coding narratives as sequences, calculating pairwise dissimilarities between sequences, and then using them to perform some form of data reduction, commonly through clustering.

2.1 Overview and basic concepts of dissimilarity-based approaches

Measuring the distance between sequences is the common starting point for all dissimilarity-based methods. The main idea is to select a way of quantifying how dissimilar two sequences are, and then apply this measure of dissimilarity to all pairs of sequences in a given dataset. From a sociological perspective, this approach aims to quantify the extent to which two individuals followed similar trajectories.

The most widely used method to measure pairwise dissimilarities of sequences is called **Optimal Matching** (OM). The general idea of OM is to measure the distance between two sequences by computing the minimal cost required to transform one of them into an exact copy of the other one, using two kinds of edit operations: substituting one state for another (**substitution**), and inserting or deleting (**indel**) a state. Each of these operations is assigned a specific cost. These costs can be state-dependent, offering high-flexibility for the OM method. Thus, OM lies between the sum of 2 terms: a weighted sum of time shifts (indels) and a weighted sum of the mismatches (substitutions) remaining after the time shifts. Consequently, sequences that are more similar will naturally have smaller distances. The OM distance can be thought of as based on the longest partially matched subsequence. From a sociological point of view, these partially matched subsequences can be interpreted as a “common narrative” between trajectories.

Although often referred to as “OM analysis”, a dissimilarity-based analysis is not limited to OM distances alone. Over time, a variety of distance measures have emerged. As extensively reviewed by Studer et al. (2015) [2], users of sequence analysis now have a wide range of distance measures to choose from. Even within the OM approach itself, many variants exist, each characterized by different methods for assigning costs to edit operations.

State sequences are complex objects that provide many different pieces of information. It is therefore highly important to select an appropriate dissimilarity measure, as different measures are more or less sensitive to different sequence characteristics. This includes the sequencing, the duration and the timing of visited states, as detailed in the work of Studer et al. (2015) [2]. Consequently, two sequences might appear very similar under one distance measure while seeming quite different under another. In theory, one should carefully, yet arbitrarily, choose the measure that aligns best with one’s analysis objectives.

After calculating the dissimilarity for all pairs of sequences, the next step typically involves cluster analysis, often performed using hierarchical clustering based on the computed distances. This process helps identify distinct "types of patterns" and partitions the sequences into groups that are as homogeneous as possible within each group and as distinct as possible between different groups.

2.2 Weaknesses of such approaches

While widely used, methods based on dissimilarity measures, such as Optimal Matching, have several weaknesses and limitations.

One of the principal weaknesses of Optimal Matching lies in the issue of appropriately defining costs for insertions, deletions, and substitutions. This point raises questions about the rigour of the approach, as it involves assigning arbitrary costs to edit operations prior to the implementation. Such arbitrary costs can lead to subjectivity in the analysis, potentially impacting the reliability of the results. Most importantly, the choice of these costs can influence the clustering outcomes, leading to different possible interpretations of the data, and making it a less consistent or robust approach. Some measures have had variants where the costs are somewhat dependent on the data, but there still remains a significant amount of arbitrariness. Furthermore, as explained previously, two sequences can be judged very similar or very different according to what dissimilarity measure has been used. So there is another layer of arbitrariness, which lies in the choice of the dissimilarity measure to apply. This is a choice that can be difficult, as there is no optimal dissimilarity measure and that the focus of a sequence analysis is typically multicriterial. This choice, once again, can heavily affect the results of the analysis and contribute to a lack of robustness.

Moreover, many existing dissimilarity measures have been criticised for the high number of parameters that must be set by the user, which can be seen as overparameterisation. This issue is further amplified by the fact that usually, as mentioned earlier, these parameters are set arbitrarily.

As explained by Liao et al. in "Sequence analysis: Its past, present, and future" (2022) [3], measures like Optimal Matching have also been highly criticised for the difficulty to sociologically interpret substitution and indel operations, as well as their associated costs. While such edit operations make more sense when applied to biology (DNA mutations. . .) or to computer science (signal changes. . .), in the social sciences, sequences can hardly be considered as the result of a transformation of another sequence.

One could also argue that another issue is the lack of interpretability of the outcomes. When employing dissimilarity-based approaches, the resulting information primarily consists of the assignments of sequences to their respective clusters. Aside from visualisation tools to separately examine these groups (often carried out on R via the `TraMineR` package, well known in sequence analysis), there are few analytical tools or insights available for further exploration.

Finally, a last weakness worth pointing out has to do with the practical implementation of these methods. When computing pairwise dissimilarities between sequences in a given dataset, the algorithms do not know how to handle cases where not all sequences are of equal length. This issue is particularly constraining when observations are not collected over the same time span. In such cases, a common practice among sequence analysis users is to introduce an artificial state to extend shorter sequences, thereby creating a dataset where all sequences have the same length. However, this approach adds extra preprocessing work and once again reduces the rigour and practicality of the method.

3 Mixtures of Semi-Markov Chains: Presentation of the Theoretical Framework

In the following section, we present the theoretical framework of the probabilistic model proposed in this work.

3.1 Semi-Markov chains

A **semi-Markov chain** is a type of stochastic model that extends the traditional Markov chain concept by introducing a more general mechanism for state transitions. It does so by considering that the chain spends a certain amount of time in each successive visited state, known as **sojourn time**, that follows an arbitrary distribution. This approach provides greater flexibility in modeling the duration of time spent in each state compared to standard Markov chains.

In this work, we focus on the case where longitudinal data is described by **categorical variables**. Consequently, the state space for the semi-Markov chains in our model can be represented as a discrete and finite set.
Furthermore, we can limit our study to chains (rather than processes) because, in the context of life trajectories and other sociological studies, the observed sequences are usually documented in discrete time.

A more rigorous definition holds as follows:

Consider $(X_n)_{n\geq 1}$ a Markov chain taking values in $S = \{1, ..., E\}$ a finite state space ($E < +\infty$), with transition matrix P.

The chain $(X_n)_{n\geq 1}$ therefore satisfies the Markov property: $\forall n \geq 1$, $\forall x_0, \ldots, x_{n-1}, x, y \in S$,

$$\begin{aligned}\mathbb{P}(X_{n+1} = y \mid X_0 = x_0, \ldots, X_{n-1} = x_{n-1}, X_n = x) &= \mathbb{P}(X_{n+1} = y \mid X_n = x) \\ &= \mathrm{P}_{xy}\end{aligned}$$

In other words, the probability of transitioning to the next state y depends only on the current state x and not on the previous states, and is stored in the matrix P.

Let us denote by $\alpha = (\alpha_1, \ldots, \alpha_E)$ the vector of initial probabilities of the Markov chain: $\forall j \in S,\ \alpha_j = \mathbb{P}(X_1 = j)$.

Let us also consider a random sequence $(B_n)_{n\geq 1}$ made by the successive sojourn times in the visited states, taking values in $T = \mathbb{N}^*$ if we are in discrete time (or in $T =]0, +\infty[$ if we are in continuous time). For all states i and j, the time spent in state i before transitioning to state j is independently distributed according to a pre-defined probability distribution. We denote by Ψ_{ij} the cumulative distribution function given the current state i and the upcoming state j of X:

$$\forall n \geq 1,\ \forall i, j \in S,\ \forall t \in T,\quad \Psi_{ij}(t) = \mathbb{P}(B_n \leq t \mid X_n = i,\ X_{n+1} = j)$$

We only consider parametric distributions for B, with a finite number of parameters.

Combined, we consider that (X_n, B_n) satisfies: $\forall n \geq 1,\ \forall i, j \in S,\ \forall t \in T$,

$$\mathbb{P}\left(X_{n+1} = j,\ B_n \leq t \mid X_n = i, X_{n-1}, \ldots, X_1, B_{n-1}, \ldots, B_1\right) = \mathrm{P}_{ij}\Psi_{ij}(t)$$

$(X_n, B_n)_{n\geq 1}$, as defined above, is called a **semi-Markov chain** on the state space S.

Its distribution is fully characterized by the set of parameters $\boldsymbol{\theta} = (\alpha,\ \mathrm{P},\ \Psi_{ij}\ ;\ i,j \in S)$.

From an interpretive perspective, we consider that :
- X_n represents the state of the chain at the n-th transition.
- B_n represents the sojourn time spent in X_n.

3.2 Finite mixture distribution

A mixture distribution is a probabilistic model that combines multiple individual distributions. The overall distribution is then a weighted sum of each of these component distributions.
It is particularly useful when the observed data exhibits heterogeneity that cannot be adequately captured by a single distribution, and where observations are believed to come from different latent subpopulations, each of which is governed by its own distribution.

Here, we consider that we have G independent groups of semi-Markov chains that all take their values in a same state space S. Each sequence is assumed to belong to one of these components, but this assignment is not directly observed. By denoting by $\pi_g > 0$ (for $g \in \{1, \ldots, G\}$) the probability of observing a Markov chain with parameters $\theta_g = (\alpha^g,\ \mathrm{P}^g,\ \Psi_{ij}^g\ ;\ i,j \in S)$, we get that the distribution of the finite mixture chain $(X_n^\pi, B_n^\pi)_{n>1}$ of the model we consider is defined by:

$$\sum_{g=1}^{G} \pi_g \cdot Law(\alpha^g,\ \mathrm{P}^g,\ \Psi_{ij}^g\ ;\ i,j \in S)$$

where:

- $Law(\alpha^g,\ \mathrm{P}^g,\ \Phi_{ij}^g\ ;\ i,j \in S)$ is the probability law of a semi-Markov chain of parameters $\theta_g = (\alpha^g,\ \mathrm{P}^g,\ \Psi_{ij}^g\ ;\ i,j \in S)$.
- π_g are the weights of the mixture components, where each weight represents the proportion of the sequences belonging to that component. These weights must verify the following conditions:

$$\sum_{g=1}^{G} \pi_g = 1 \quad \text{and} \quad \pi_g \geq 0 \quad \text{for all } g.$$

3.3 Theoretical framework and specific case of the model

In our model, we establish that the sojourn times follow **gamma distributions**, due to their adaptability, simple moments, and their capability to fit a wide range of sojourn time shapes and patterns. For all n, B_n follows a gamma distribution described by 2 parameters, a shape $a > 0$ and a rate $\lambda > 0$, whose density is given by: for $x \geq 0$,

$$\Phi(x; a, \lambda) = \frac{\lambda^a x^{a-1} e^{-\lambda x}}{\Gamma(a)}$$

where Γ is the Gamma function:

$$\Gamma(a) = \int_0^\infty t^{a-1} e^{-t}\, dt\ , \quad \forall a > 0$$

The expectation of the gamma distribution is given by a/λ and its variance by a/λ^2.

We also assume that there is **no anticipation**, in the sense that the considered sojourn time distribution only depends on the current state, and no longer on the next state. This assumption seems relevant from a sociological perspective as, instinctively, one doesn't know how long one remains in a certain state before going to the next, or at least the time spent in it does not depend on the upcoming state, since the future is uncertain. Additionally, it significantly reduces the number of parameters to estimate (instead of requiring $2GE(E-1)$ parameters to characterize the sojourn times, we only need $2GE$).

The set of parameters that characterizes a semi-Markov chain is now described by $\theta = (\alpha,\ \mathrm{P},\ a_j, \lambda_j\ ;\ j \in S)$.

Here, the core of the objective is to model life trajectories and other sociological longitudinal data as a mixture of semi-Markov chains. For this purpose, we can consider sequences as being transitions from one state in each visited state. As an example, we consider the longitudinal data sequences presented in Table 1. Each row in the table represents a sequence of observations for an individual over several time points. Each observed state belongs to the state space $S = \{1, 2, 3\}$.

Table 1: Example of sequences of longitudinal data

Individual	Time 1	Time 2	Time 3	Time 4	Time 5	Time 6	Time 7	Time 8
1	2	2	3	1	1	1	1	1
2	1	1	1	3	3	3	3	3
3	3	1	1	1	1	3	3	2

For each individual, we can describe their sequence as a semi-Markov chain $(X_n, B_n)_n$, with X the states and B the sojourn times, on the state space $S = \{1, 2, 3\}$. More specifically, we have:

- Individual 1: $(X_n^1, B_n^1)_n$

Sequence 1 = (2, 2, 3, 1, 1, 1, 1, 1)

n	1	2	3
X_n^1	**2**	**3**	**1**
B_n^1	**2**	**1**	**5**

- Individual 2: $(X_n^2, B_n^2)_n$

Sequence 2 = (1, 1, 1, 3, 3, 3, 3, 3)

n	1	2
X_n^2	**1**	**3**
B_n^2	**3**	**5**

- Individual 3: $(X_n^3, B_n^3)_n$

Sequence 3 = (3, 1, 1, 1, 1, 3, 3, 2)

n	1	2	3	4
X_n^3	**3**	**1**	**3**	**2**
B_n^3	**1**	**4**	**2**	**1**

Moreover, it has often been suggested that individuals form non-homogeneous populations, so modeling the sequences with the help of a mixture model is of real interest. By doing so, in a sociological context, we are able to extract sub-populations, more commonly known as **clusters**, and we are thus in a position to interpret the parameters of the semi-Markov chains associated to each cluster in order to study separate behaviours.

4 Algorithm and Parameter Estimation

In this section, we present the algorithm that we have implemented to identify clusters within a given input dataset and estimate the corresponding parameters for each cluster. This algorithm not only allows for the partitioning of data into meaningful groups but also determines the optimal number of clusters. Additionally, it iteratively refines the parameter estimates for the semi-Markov model introduced in the previous section, ensuring both accurate clustering and robust parameter estimation.

4.1 Basic concepts and overview

Consider a sample of n independent sequences $(S_i)_{i\in\{1,\dots,n\}}$. We assume that the $S_1,\dots,S_n$ are drawn from a mixture of a certain number $\boldsymbol{G}$ of semi-Markov chains.

The algorithm presented in the following section is capable of handling sequences of unequal lengths, whether in terms of the total number of visited states or the duration over which the observations are collected. We denote by N_i, for all i, the number of states visited by sequence S_i. We thus have, for $i \in \{1,\dots,n\}$:

$$S_i = (X_1^i, B_1^i, \dots, X_{N_i-1}^i, B_{N_i-1}^i, X_{N_i}^i, B_{N_i}^i)$$

belonging to a certain group $g \in \{1,\dots,G\}$ of the mixture.

For the moment, we suppose that the number of mixture components G is known, but we will later come back to the issue of determining its optimal value.

The goal here is first to obtain an estimation of the set of all the parameters that characterise the mixture of semi-Markov chains associated to the set of sequences: $\boldsymbol{\theta} = (\pi, \theta_1, \dots, \theta_G)$, where $\pi = (\pi_1, \dots, \pi_G)$ is the vector of probabilities of belonging to each mixture component and $\theta_g = (\alpha^g,\ \mathrm{P}^g,\ \Phi_{ij}^g\ ;\ i,j \in S)$ for all $g \in \{1,\dots,G\}$), as well as obtaining the assignment information of what sequence belongs to what mixture component.

For this matter, we apply the **Expectation-Maximisation (EM)** method, commonly used for determining the parameters of mixture distributions, based on likelihood maximisation. Then, a **MAP** criterion is applied to assign each individual sequence to one of the clusters, and finally, the **BIC** criterion is employed to select the optimal number of groups G.

Before delving into the details of the algorithm, one has to make sure that the issue of identifiability is taken care of. Identifiability refers to the ability of the algorithm to uniquely estimate the model parameters, based on the data. It is a very important condition to ensure the convergence of estimation algorithms to a unique solution.

We thus impose the 2 following constraints:

- The observation period of the $(S_i)_i$ must span over at least 4 time points.
  Hence, must be verified: $\sum_{i=1}^{N_i} B_i \geq 4$, for all $i \in \{1,\dots,n\}$.

- The number of groups in the mixture must not be too large compared to the number of states. More precisely, one must have: $2G \leq E$, where G is the number of mixture components and E is the total number of states.

Additionnally, it is important to note that for the algorithm to interpret the state space, it must be represented in the form $S = \{1, \ldots, E\}$. Therefore, if categorical variables are involved, one must first "translate" them to numerical values ranging from 1 to E, in order to be compatible with the algorithm.

4.2 EM algorithm

The Expectation-Maximisation (EM) algorithm is a widely used iterative method for finding maximum likelihood estimates of parameters in statistical models. It is particularly useful in contexts like ours where we need to estimate parameters from a mixture model, where the mixture component assignments are unknown.
The approach is to consider that the unknown mixture components assignments are missing observations, or hidden variables, denoted by $(Z_i)_{i\in\{1,\ldots,n\}}$. We regard them as the missing classifying indicators, where each Z_i is a vector of G elements, with a single one and $G-1$ zeros: if S_i is derived from the g-th mixture group, then $Z_{ig} = 1$ and $Z_{ik} = 0 \ \ \forall k \neq g$.

The general idea is that at every iteration, we compute the expected log-likelihood based on what was estimated during the previous iteration (E-step), expected log-likelihood that we then maximise with respect to the varying parameters in order to update their estimates (M-step).

4.2.1 Likelihood of the model

A large part of the EM algorithm is based on the concept of maximum of likelihood estimators. We introduce here the log-likelihood of the model given the observed sequences and the "current" parameter estimates during a given iteration.

Let us denote by Φ_j^g the gamma density function with parameters (a_j^g, λ_j^g) , $j \in S$, $g \in \{1,...,G\}$.
For a given sequence S_i, $i \in \{1,...,n\}$, belonging to a given group $g \in \{1,...,G\}$, its log-likelihood is described by:

$$\begin{aligned}\ln \mathcal{L}_g(\theta_g; S_i) &= \ln\left(\alpha^g_{X_1^i} \cdot \Phi^g_{X_1^i}(B_1^i) \cdot \prod_{k=2}^{N_i} P^g_{X_{k-1}^i X_k^i} \cdot \Phi^g_{X_k^i}(B_k^i)\right) \\ &= \ln(\alpha^g_{X_1^i}) + \sum_{k=1}^{N_i} \ln(\Phi^g_{X_k^i}(B_k^i)) + \sum_{k=2}^{N_i} \ln(P^g_{X_{k-1}^i X_k^i}) \qquad (1)\end{aligned}$$

The log-likelihood of the entire set of sequences is given by:

$$\ln \mathcal{L}(\theta; S_1, \ldots, S_n) = \sum_{i=1}^{n} \ \ln(\sum_{g=1}^{G} \pi_g \ \mathcal{L}_g(\theta_g; S_i) \ ) \qquad (2)$$

This expression is however difficult to use in the next steps, so we inject the indicators $(Z_i)_{i\in\{1,\ldots,n\}}$ defined earlier to simplify its implementation. We thus have:

$$\ln \mathcal{L}(\theta; S_1, Z_1, \ldots, S_n, Z_n) = \sum_{i=1}^{n}\sum_{g=1}^{G} \ Z_{ig} \ln(\pi_g \ \mathcal{L}_g(\theta_g; S_i)) \qquad (3)$$

4.2.2 Step E

The Expectation step essentially involves estimating what the missing data might be, namely the $(Z_i)_i$, based on the current parameter values, and using it to derive the expected log-likelihood. We assume here that we are at a specific iteration m and that an estimate of the parameters has been calculated during the previous iteration, denoted by $\theta^{(m-1)}$.

For all $i \in \{1,...,n\}$ and $g \in \{1,...,G\}$, we estimate $\hat{Z}_{ig}^{(m)}$ as the conditional probability that the sequence S_i has been generated by the mixture component g, using the parameters $\theta^{(m-1)}$. We obtain, with the Bayes theorem :

$$\begin{aligned}
\boxed{\hat{Z}_{ig}^{(m)}} &= \mathbb{E}[Z_{ig} \mid S_i;\ \theta^{(m-1)}] \\
&= \mathbb{P}(Z_{ig} = 1 \mid S_i;\ \theta^{(m-1)}) \\
&= \boxed{\frac{\pi_g^{(m-1)} \mathcal{L}_g(\theta^{(m-1)}; S_i)}{\sum_{j=1}^{G} \pi_j^{(m-1)} \mathcal{L}_j(\theta^{(m-1)}; S_i)}}
\end{aligned}$$

We can now write the **expected log-likelihood** of the complete data given $\theta^{(m-1)}$ and $\hat{Z}_{ig}^{(m)}$, for all i and g:

$$\begin{aligned}
A(\theta; \theta^{(m-1)}) &= \mathbb{E}[\ \ln \mathcal{L}(\theta; S_1, \hat{Z}_1^{(m)}, \dots, S_n, \hat{Z}_n^{(m)}) \mid S_1, \dots, S_n, \theta^{(m-1)}\ ] \\
&= \mathbb{E}[\ \sum_{i=1}^{n} \sum_{g=1}^{G} \hat{Z}_{ig}^{(m)} \ln(\pi_g \mathcal{L}_g(\theta_g; S_i) |\ S_1, \dots, S_n, \theta^{(m-1)}\ ] \\
&= \sum_{i=1}^{n} \sum_{g=1}^{G} \mathbb{E}[\hat{Z}_{ig}^{(m)} \ln(\pi_g) \mid S_1,..., S_n, \theta^{(m-1)}\ ] + \sum_{i=1}^{n} \sum_{g=1}^{G} \mathbb{E}[\hat{Z}_{ig}^{(m)} \mathcal{L}_g(\theta_g; S_i) \mid S_1,..., S_n, \theta^{(m-1)}\ ] \\
&= \sum_{i=1}^{n} \sum_{g=1}^{G} \hat{Z}_{ig}^{(m)} \ln(\pi_g^{(m-1)}) + \sum_{i=1}^{n} \sum_{g=1}^{G} \hat{Z}_{ig}^{(m)} \ln \mathcal{L}_g(\theta_g^{(m-1)}; S_i) \\
&= \sum_{i=1}^{n} \sum_{g=1}^{G} \hat{Z}_{ig}^{(m)} \ln(\pi_g^{(m-1)}) + \sum_{i=1}^{n} \sum_{g=1}^{G} \hat{Z}_{ig}^{(m)} \ln(\alpha_{X_1^i}^{g\ (m-1)}) \qquad (4) \\
&\quad + \sum_{i=1}^{n} \sum_{g=1}^{G} \hat{Z}_{ig}^{(m)} \sum_{k=1}^{N_i} \ln(\Phi_{X_k^i}^{g\ (m-1)}(B_k^i)) + \sum_{i=1}^{n} \sum_{g=1}^{G} \hat{Z}_{ig}^{(m)} \sum_{k=2}^{N_i} \ln(\mathrm{P}_{X_{k-1}^i X_k^i}^{g\ \ (m-1)})
\end{aligned}$$

We have used the fact that, by definition, each $\hat{Z}_{ig}^{(m)}$ is $\sigma(\theta^{(m-1)}, S_i)$-measurable.

4.2.3 Step M

In the Maximisation step, we maximise the expected loglikelihood from the E-step in (4) with respect to the parameters, given the current values of the $\hat{Z}_{ig}^{(m)}$. We then use these optimal values to update the parameter estimates. In other words, we adjust the parameters to best fit the data, considering the new estimations for the mixture group assignments.

We obtain explicit formulas for updating most parameters.

Updating the mixture probabilities:

for all $g \in \{1,..., G\}$, we have:

$$\boxed{\hat{\pi}_g^{(m)} = \frac{1}{n} \sum_{i=1}^{n} \hat{Z}_{ig}^{(m)}}$$

Proof We can extract from expression (4) the only part that depends on the $\pi = (\pi_1 .., \pi_G)$ parameter:

$$A(\theta; \theta^{(m-1)}) = \ldots + \sum_{i=1}^{n} \sum_{g=1}^{G} \hat{Z}_{ig}^{(m)} \ln(\pi_g^{(m-1)}) + \ldots$$

We need to maximise this part with respect to π under the constraint $\sum_{g=1}^{G} \pi_g = 1$.
To solve this constrained optimisation problem, we use the method of Lagrange multipliers, by setting the following Lagrangian function:

$$\mathrm{L}(\pi, \lambda) = \sum_{i=1}^{n} \sum_{g=1}^{G} \hat{Z}_{ig}^{(m)} \ln(\pi_g) + \lambda \ (1 - \sum_{g=1}^{G} \pi_g), \quad \text{with } \lambda \text{ in } \mathbb{R}$$

We then solve the system of equations:

$$\begin{cases} \dfrac{\partial \mathrm{L}}{\partial \pi}(\pi, \lambda) = 0 & \text{(a)} \\ \displaystyle\sum_{g=1}^{G} \pi_g = 1 & \text{(b)} \end{cases}$$

Applying the partial derivatives for all g from 1 to G , we get from (a):

$$\text{(a)} \iff \sum_{i=1}^{n} \hat{Z}_{ig}^{(m)} \cdot \frac{1}{\pi_g} - \lambda = 0 \iff \pi_g = \frac{\sum_{i=1}^{n} \hat{Z}_{ig}^{(m)}}{\lambda}$$

Combined with (b), we get: (b) $\iff \sum_{g=1}^{G} \sum_{i=1}^{n} \frac{\hat{Z}_{ig}^{(m)}}{\lambda} = 1 \iff \lambda = \sum_{i=1}^{n} \sum_{g=1}^{G} \hat{Z}_{ig}^{(m)}$.

Since $\sum_{g=1}^{G} \hat{Z}_{ig}^{(m)} = 1$, we have: $\lambda = \sum_{i=1}^{n} 1 = n$.

Thus, the solution of the system is given by

$$\pi_g = \frac{1}{n} \sum_{i=1}^{n} \hat{Z}_{ig}^{(m)}, \quad \forall g \in \{1, \ldots, G\}$$

It is the unique solution of the problem given the objective is strictly concave. □

Updating the initial probabilities:
for all $g \in \{1,..., G\}$, and for all $j \in S$ we have:

$$\boxed{\hat{\alpha}_j^{g(m)} = \frac{\sum_{i=1}^{n} \hat{Z}_{ig}^{(m)} \ \mathbb{1}_{\{X_1^i = j\}}}{\sum_{i=1}^{n} \hat{Z}_{ig}^{(m)}}}$$

Proof Similar to the previous proof, we can extract from A the only part that depends on the α parameters:

$$A(\theta;\theta^{(m-1)}) = \ldots + \sum_{i=1}^{n}\sum_{g=1}^{G} \hat{Z}_{ig}^{(m)} \ln(\alpha_{X_1^i}^{g\ (m-1)}) + \ldots$$

which we must maximise with respect to the $\alpha = (\alpha_j^g; j \in S, g \in \{1, \ldots, G\})$ under the constraint $\sum_{j=1}^{E} \alpha_j^g = 1$, given the set of observed sequences. The Lagrangian function is thus given by:

$$\begin{aligned}
\mathrm{L}(\alpha,\lambda) &= \sum_{i=1}^{n}\sum_{g=1}^{G} \hat{Z}_{ig}^{(m)} \ln(\alpha_{X_1^i}^{g}) + \sum_{g=1}^{G} \lambda_g \ (1 - \sum_{j=1}^{E} \alpha_j^g) \\
&= \sum_{i=1}^{n}\sum_{g=1}^{G}\sum_{j=1}^{E} \hat{Z}_{ig}^{(m)} \ln(\alpha_j^g)\ \mathbb{1}_{\{X_1^i=j\}} + \sum_{g=1}^{G} \lambda_g \ (1 - \sum_{j=1}^{E} \alpha_j^g)
\end{aligned}$$

with $\lambda_1, \ldots, \lambda_G \in \mathbb{R}$.

We solve:

$$\begin{cases}
\dfrac{\partial \mathrm{L}}{\partial \alpha_j^g}\ (\alpha,\lambda) = 0,\ \forall j, \forall g & \text{(a)} \\[2ex]
\displaystyle\sum_{j=1}^{E} \alpha_j = \ 1,\ \forall g & \text{(b)}
\end{cases}$$

From (a), we get, for all j, g :

$$\text{(a)} \iff \sum_{i=1}^{n} \hat{Z}_{ig}^{(m)} \cdot \frac{1}{\alpha_j^g} \mathbb{1}_{\{X_1^i=j\}} - \lambda_g = 0 \iff \alpha_j^g = -\frac{1}{\lambda_g} \cdot \sum_{i=1}^{n} \hat{Z}_{ig}^{(m)} \mathbb{1}_{\{X_1^i=j\}}$$

Combined with (b), we have:

$$\begin{aligned}
\text{(b)} &\iff \sum_{j=1}^{E} \frac{1}{\lambda_g} \cdot \sum_{i=1}^{n} \hat{Z}_{ig}^{(m)} \mathbb{1}_{\{X_1^i=j\}} = -1 \\
&\iff -\lambda_g = \sum_{i=1}^{n} \hat{Z}_{ig}^{(m)} \sum_{j=1}^{E} \mathbb{1}_{\{X_1^i=j\}} \\
&\iff -\lambda_g = \sum_{i=1}^{n} \hat{Z}_{ig}^{(m)} \quad \text{because} \ \sum_{j=1}^{E} \mathbb{1}_{\{X_1^i=j\}} = 1
\end{aligned}$$

Thus, we obtain the solution: $\alpha_j^g = \dfrac{\sum_{i=1}^{n} \hat{Z}_{ig}^{(m)} \mathbb{1}_{\{X_1^i=j\}}}{\sum_{i=1}^{n} \hat{Z}_{ig}^{(m)}}$, for all j and all g, unique solution as the objective is strictly concave. □

Updating the transition matrices:
for all $g \in \{1,...,G\}$, and for all $h,\ j \in S$ we have:

$$\boxed{\hat{\mathrm{P}}_{hj}^{g\ (m)} = \frac{\sum_{i=1}^{n} \hat{Z}_{ig}^{(m)}\ n_{hj}^{i}}{\sum_{l=1}^{E}\sum_{i=1}^{n} \hat{Z}_{ig}^{(m)} n_{hl}^{i}}}$$

where n^i_{hj} is the number of transitions going from state h to state j in sequence S_i.

Proof Again, we can extract from A the only part the depends on the P parameters:

$$A(\theta;\theta^{(m-1)}) = \ldots + \sum_{i=1}^{n}\sum_{g=1}^{G}\hat{Z}^{(m)}_{ig}\sum_{k=2}^{N_i}\ln(\mathrm{P}^{g}_{X^i_{k-1}X^i_k}{}^{(m-1)}) + \ldots$$

which we must maximise with respect to the $\mathrm{P} = (\mathrm{P}^g_{hj};\ h,j \in S,\ g \in \{1,...,G\})$ under the constraints $\sum_{j=1}^{E}\mathrm{P}_{hj} = 1$ and $\mathrm{P}_{hh} = 0,\ \forall h \in S$, given the set of observed sequences. The Lagrangian function here is given by:

$$\begin{aligned}\mathrm{L}(\mathrm{P},\lambda,\mu) &= \sum_{i=1}^{n}\sum_{g=1}^{G}\hat{Z}^{(m)}_{ig}\sum_{k=2}^{N_i}\ln(\mathrm{P}^g_{X^i_{k-1}X^i_k}) + \sum_{g=1}^{G}\sum_{h=1}^{E}\mu^h_g\mathrm{P}^g_{hh} + \sum_{g=1}^{G}\sum_{h=1}^{E}\lambda^h_g(1-\sum_{j=1}^{E}\mathrm{P}^g_{hj})\\ &= \sum_{i=1}^{n}\sum_{g=1}^{G}\sum_{k=2}^{N_i}\sum_{h=1}^{E}\sum_{j=1}^{E}\hat{Z}^{(m)}_{ig}\ln(\mathrm{P}^g_{hj})\mathbb{1}_{\{X^i_{k-1}=h\}}\mathbb{1}_{\{X^i_k=j\}} + \sum_{g=1}^{G}\sum_{h=1}^{E}\left(\mu^h_g\mathrm{P}^g_{hh} + \lambda^h_g(1-\sum_{j=1}^{E}\mathrm{P}^g_{hj})\right)\end{aligned}$$

with $\lambda^h_g, \mu^h_g \in \mathbb{R},\ \forall h \in S,\ \forall g \in \{1,...,G\})$.

We solve:

$$\begin{cases}\dfrac{\partial \mathrm{L}}{\partial \mathrm{P}^g_{hj}}(\mathrm{P},\lambda,\mu) = 0,\ \forall h,j,\forall g & \text{(a)}\\ \mathrm{P}^g_{hh} = 0,\ \forall h,\forall g & \text{(b)}\\ \displaystyle\sum_{j=1}^{E}\mathrm{P}^g_{hj} = \ 1,\ \forall h,\forall g & \text{(c)}\end{cases}$$

From (a), we get $\forall\ h,j,g$:

$$\begin{aligned}\text{(a)} \iff& \sum_{i=1}^{n}\sum_{k=2}^{N_i}\hat{Z}^{(m)}_{ig}\ \frac{1}{\mathrm{P}^g_{hj}}\ \mathbb{1}_{\{X^i_{k-1}=h\}}\mathbb{1}_{\{X^i_k=j\}} - \lambda^h_g = 0\\ \iff& \mathrm{P}^g_{hj} = \frac{1}{\lambda^h_g}\sum_{i=1}^{n}\hat{Z}^{(m)}_{ig}\sum_{k=2}^{N_i}\mathbb{1}_{\{X^i_{k-1}=h\}}\mathbb{1}_{\{X^i_k=j\}}\\ \iff& \mathrm{P}^g_{hj} = \frac{1}{\lambda^h_g}\sum_{i=1}^{n}\hat{Z}^{(m)}_{ig}\ n^i_{hj}\end{aligned}$$

because, by definition, $n^i_{hj} = \sum_{k=2}^{N_i}\mathbb{1}_{\{X^i_{k-1}=h\}}\mathbb{1}_{\{X^i_k=j\}}$.

Combined with (c), we have:

$$\begin{aligned}\text{(c)} \iff& \sum_{j=1}^{E}\frac{1}{\lambda^h_g}\sum_{i=1}^{n}\hat{Z}^{(m)}_{ig}\ n^i_{hj} = 1\\ \iff& \lambda^h_g = \sum_{j=1}^{E}\sum_{i=1}^{n}\hat{Z}^{(m)}_{ig}n^i_{hj}\end{aligned}$$

Thus, we finally obtain the solution, for all g and all h, j: $\mathrm{P}^g_{hj} = \dfrac{\sum_{i=1}^n \hat{Z}_{ig}^{(m)} n^i_{hj}}{\sum_{l=1}^E \sum_{i=1}^n \hat{Z}_{ig}^{(m)} n^i_{jl}}$, unique solution as the objective is strictly concave.

□

Updating the gamma distributions parameters:
When it comes to the parameters of the gamma distributions, we do not have an explicit solution for the maximisation of:

$$A(\theta; \theta^{(m-1)}) = \ldots + \sum_{i=1}^{n} \sum_{g=1}^{G} \hat{Z}_{ig}^{(m)} \sum_{k=1}^{N_i} \ln(\Phi^{g\ (m-1)}_{X^i_k}(B^i_k)) + \ldots$$

with respect to the $((a^g_j, \lambda^g_j);\ j \in S)_{g \in \{1,\ldots,G\}}$, where Φ is the density function of the gamma distribution.
We thus estimate the maximum by using classical numerical optimisation algorithms, such as the `BFGS` method, a robust quasi-Newton method.
However, as explained in Cardot et al. (2018) [1], the log likelihood is not bounded in the gamma mixture models and adding a penalty is required to ensure the consistency of maximum likelihood estimators, which acts on the shape parameter a of the gamma distribution.
The penalty we apply is defined as:

$$Pen\,(a^g_l,\ \forall l \in S,\ \forall g \in \{1,...,G\}) = -\frac{1}{\sqrt{\sum_{i=1}^n N_i}} \sum_{g=1}^{G} \sum_{l=1}^{E} (a^g_l + \ln a^g_l)$$

Hence, the function to be maximised is defined as follows:

$$\begin{aligned}
f((a^g_l, \lambda^g_l);\ l \in S, g \in \{1,...,G\}) &= \sum_{i=1}^{n} \sum_{g=1}^{G} \hat{Z}_{ig}^{(m)} \sum_{k=1}^{N_i} \ln(\Phi^g_{X^i_k}(B^i_k)) + Pen(a^g_l;\ \forall l, g) \\
&= \sum_{i=1}^{n} \sum_{g=1}^{G} \hat{Z}_{ig}^{(m)} \sum_{k=1}^{N_i} \ln\left(\prod_{l=1}^{E} \Phi_l^{g \cdot \mathbb{1}_{\{X^i_k = l\}}}(B^i_k) \right) + Pen(a^g_l;\ \forall l, g) \\
&= \sum_{i=1}^{n} \sum_{g=1}^{G} \hat{Z}_{ig}^{(m)} \sum_{k=1}^{N_i} \sum_{l=1}^{E} \mathbb{1}_{\{X^i_k = l\}} \ln(\Phi^g_l(B^i_k)) + Pen(a^g_l;\ \forall l, g) \\
&= \sum_{l=1}^{E} \sum_{g=1}^{G} \left(\sum_{i=1}^{n} \sum_{k=1}^{N_i} \hat{Z}_{ig}^{(m)} \mathbb{1}_{\{X^i_k = l\}} \ln(\Phi^g_l(B^i_k)) \right) + Pen(a^g_l;\ \forall l, g)
\end{aligned}$$

For all state l and for all group g, we can factorise this function, so all gamma distribution parameters (a^g_l, λ^g_l) for a fixed l and g can be estimated by applying the numerical optimisation procedure on the function:

$$\boxed{f_{l,g}(a^g_l, \lambda^g_l) = \sum_{i=1}^{n} \sum_{k=1}^{N_i} \hat{Z}_{ig}^{(m)} \mathbb{1}_{\{X^i_k = l\}} \ln(\Phi^g_l(B^i_k)) - \frac{a^g_l + \ln(a^g_l)}{\sqrt{\sum_{i=1}^n N_i}}}$$

$$\boxed{(\hat{a}^g_l, \hat{\lambda}^g_l)^{(m)} = \max f_{l,g}(a^g_l, \lambda^g_l)}$$

where Φ^g_l is the density function for the gamma distribution with parameters (a^g_l, λ^g_l), and with the lower bound constraints: $a^g_l > 0$ and $\lambda^g_l > 0$ for all $l \in S$ and for all $g \in \{1, \ldots, G\}$.

4.2.4 Initialisation

As for many iterative approaches, the result (as well as the number of required iterations) of the EM algorithm depends heavily on the starting points assigned to the various parameters, given that every estimation $\hat{\theta}^{(m)}$ depends on the previous $\hat{\theta}^{(m-1)}$. Hence, an initial value $\hat{\theta}^{(0)}$ must be carefully chosen at the beginning.

The first step is to extract the mean sojourn times spent in each possible state in S for every sequence in the dataset. Then, one performs a classic Hartigan and Wong k-means algorithm with G classes, based on these mean sojourn times per state and per individual. One should obtain a list of sequences for each resulting cluster — it is from this classification that we derive the initialisation values.

Here is how we then proceed for each parameter initialisation:

- The weights ($\pi^g;\ g \in \{1,...,G\}$):
  We initialise them uniformly, by setting $\hat{\pi}_g^{(0)} = \frac{1}{G}$ for all $g \in \{1,\dots,G\}$.
- The initial probabilities ($\alpha_1^g, \dots, \alpha_E^g;\ g \in \{1,...,G\}$):
  We use the k-means results and for every cluster and every state, we calculate the empirical proportion of sequences whose first visited state is the state of interest.
  We have, for all $j \in S$ and all $g \in \{1,...,G\}$:

  $$\hat{\alpha}_j^{g\,(0)} = \frac{\sum_{i=1}^{n} \mathbb{1}_{\{X_1^i = j\}}\ \mathbb{1}_{\{X^i \in g\}}}{\sum_{i=1}^{n}\ \mathbb{1}_{\{X^i \in g\}}}$$

- The transition matrices ($\mathrm{P}_{hj}^g;\ h,j \in S,\ g \in \{1,...,G\}$):
  Similarly to the initial probabilities, for each pair of states (h,j), and for each cluster that we have obtained through the k-means algorithm, we calculate the empirical proportion of transitions going from state h to j among all the transitions leaving state h.
  Mathematically, for all states $h,j \in S$ and all $g \in \{1,\dots,G\}$, we have:

  $$\hat{\mathrm{P}}_{hj}^{g\,(0)} = \frac{\sum_{i=1}^{n} \sum_{k=1}^{N_i-1}\ \mathbb{1}_{\{X_k^i = h\}} \mathbb{1}_{\{X_{k+1}^i = j\}}\ \mathbb{1}_{\{X^i \in g\}}}{\sum_{i=1}^{n} \sum_{k=1}^{N_i-1}\ \mathbb{1}_{\{X_k^i = h\}}\ \mathbb{1}_{\{X^i \in g\}}}$$

- The parameters for the gamma distributions ($(a_j^g, \lambda_j^g);\ j \in S,\ g \in \{1,...,G\}$):
  Here, we use the method of moments by taking advantage of the simple gamma distribution moments. We first need to extract the empirical mean μ and variance σ^2 of the sojourn times for each specific state for each cluster obtained with the k-means algorithm.
  We have:

  $$\begin{cases} \mu = \frac{a}{\lambda} \\ \sigma^2 = \frac{a}{\lambda^2} \end{cases} \iff \begin{cases} a = \frac{\mu^2}{\sigma^2} \\ \lambda = \frac{\mu}{\sigma^2} \end{cases}$$

  We can thus apply these formulas to all specific state j and cluster g, with mean μ_j^g and variance $\sigma^{2}{}_j^g$ of sojourn times.
  There is one problematic case, which is when the derived variance is zero: $\sigma^{2}{}_j^g = 0$. Either the state is not visited at all by the sequences in the given cluster, or the state has only been visited a single time. The solution is either to set the variance to `NA` in the first case, or to an arbitrary (and temporary) value in the latter.

4.3 Final cluster assignment with MAP criterion

Once the algorithm has converged, or has reached the maximum number of iterations, suppose at iteration M, we use the $\hat{Z}_{ig}^{(M)}$ ($\forall i \in \{1,...,n\}, \forall g \in \{1,...,G\}$) to definitely assign each sequence to the mixture component group it belongs to, by means of a **Maximum A Posteriori** (MAP) criterion:

$$\hat{Z}_{ig} = \text{MAP}(\hat{Z}_{ig}^{(M)}) = \begin{cases} 1 & \text{if } g = \arg\max_h(\hat{Z}_{ih}^{(M)}) \\ 0 & \text{otherwise} \end{cases}$$

4.4 Selection of the number of mixture components with BIC criterion

In practice, we usually do not know the number of clusters there is in the data, or at least, we do not know what its most appropriate number is. To address this issue, we employ the **Bayesian Information Criterion** (BIC).
This criterion is widely used for model selection, by providing a compromise between the goodness-of-fit to the data and model complexity. It helps to prevent overfitting by penalizing model complexity, thus guiding the selection towards models that achieve a good fit with a reasonable number of parameters. We use it here to determine the optimal number of components in the mixture model.

The Bayesian Information Criterion is defined as:

$$\text{BIC} = q \ln(n) - 2\ln(\hat{\text{L}}),$$

where:

- n is the number of observed sequences.

- $\ln(\hat{\text{L}}) = \ln \mathcal{L}(\hat{\theta}(G);\ S_1, \ldots, S_n)$ is the maximum loglikelihood of the model, as defined in (2) with the estimated parameters $\hat{\theta}(G)$ resulting from the EM-algorithm when G in the number of components in the mixture.

- q is the number of parameters in the model. In the framework of our model, more precisely in the particular case of gamma distributions for the sojourn times with no anticipation, we have: $q = GE(E+1) - 1$, with E the total number of states.

  Proof We have $q = \text{card}(\theta(G))$,
  where $\theta(G) = \Big((\pi_g), (\alpha_j^g)_{j\in\{1,\ldots,E\}}, (\text{P}_{hj}^g)_{h,j\in\{1,\ldots,E\}}, ((a_j,\lambda_j)^g)_{j\in\{1,\ldots,E\}}\Big)_{g\in\{1,\ldots,G\}}$.
  So

$$\begin{aligned} q &= \text{card}[(\pi_g)_g] + \text{card}[(\alpha_j^g)_{g,j}] + \text{card}[(\text{P}_{hj}^g)_{g,h,j}] + \text{card}[(((a_j,\lambda_j)^g))_{g,j}] \\ &= (G-1) + G(E-1) + G(D(D-2)) + G(2D) \\ &= G(1 + E - 1 + E(E-2) + 2E) - 1 \\ &= GE(E+1) - 1 \end{aligned}$$

  We have used the fact that the following constraints are imposed upon the parameters:

$$\sum_{g=1}^{G} \pi_g = 1 \ ; \quad \sum_{j=1}^{E} \alpha_j = 1 \ ; \quad \forall h \in S,\ \sum_{j=1}^{E} \text{P}_{hj} = 1 \text{ and } \text{P}_{hh} = 0$$

□

A lower BIC value indicates a better model fit with fewer parameters. To select the number of components G in the mixture model, we fit models with different numbers G of components and compute the BIC for each model. The model with the **lowest BIC** is considered to be the optimal choice.

Remember that for identifiability reasons, one must keep having $2G \leq E$.

It is however necessary to mention that the BIC has good asymptotic properties, so in the case where the number of observed sequences is relatively low, it is advised to keep a critical eye on the result offered by the BIC and to look at the algorithm outputs in more detail (or perhaps to arbitrarily choose a different number of clusters if the BIC for a different value of G is close enough to the one chosen by the criterion).

5 Simulation study

Assessing the efficiency and accuracy of the algorithm is challenging due to the unsupervised nature of clustering, which does not provide reference values for comparison. To overcome this, we carry out a simulation study by applying the algorithm to sequences generated from semi-Markov chains with known parameters. This approach enables us to directly compare the estimated parameters with the original, known parameters, thereby assessing the model's performance.

5.1 Simulation framework

For our simulation study, we have considered the state space $S = \{1, 2, 3, 4\}$ and a mixture made of $G = 2$ groups, holding the parameters given in Table 2, with the mixture probabilities being uniform: $\pi = (1/2,\ 1/2)$.

Table 2: Parameters for the simulation study

	Group 1	**Group 2**
Initial Probabilities:	$\alpha^1 = \begin{pmatrix} 0.5 \\ 0.1 \\ 0.1 \\ 0.3 \end{pmatrix}$	$\alpha^2 = \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}$
Transition Matrix:	$\mathrm{P}^1 = \begin{pmatrix} 0 & 0.8 & 0.1 & 0.1 \\ 0.3 & 0 & 0.3 & 0.3 \\ 0.95 & 0.05 & 0 & 0 \\ 0.4 & 0.4 & 0.2 & 0 \end{pmatrix}$	$\mathrm{P}^2 = \begin{pmatrix} 0 & 0.2 & 0.3 & 0.5 \\ 0.1 & 0 & 0.3 & 0.6 \\ 0.05 & 0.05 & 0 & 0.9 \\ 0.3 & 0.3 & 0.4 & 0 \end{pmatrix}$
Gamma Parameters:	$\Gamma_{a^1,\lambda^1} = \begin{pmatrix} 2 & 1 \\ 3 & 2 \\ 1 & 1 \\ 4 & 2 \end{pmatrix}$	$\Gamma_{a^2,\lambda^2} = \begin{pmatrix} 4 & 4 \\ 3 & 4 \\ 2 & 2 \\ 4 & 2 \end{pmatrix}$

We have simulated data for different sample sizes (100, 250, and 500) and various sequence lengths within each sample (3–10, 10–20, and 20–30). By "sequence length", we refer to the number of consecutive distinct states visited in a sequence. The length of each simulated sequence has been uniformly selected within its associated range of possible lengths. This makes a total of 9 different configurations for the simulated data, all being based on the same original parameters, described in Table 2.

For each of these configurations, 100 simulated datasets have been generated, to ensure a reliable assessment of the output's performance. In every set of simulations, we apply the algorithm and are returned: the estimated parameters, the assignment to the clusters, and the optimal number of groups chosen through BIC criterion.

We define the convergence of the algorithm as the point where the difference in model likelihood (as defined in (3)) between successive iterations falls bellow a specified tolerance, that we have set here to 10^{-5}.
We have also set the maximum number of iterations to 50, although the algorithm typically converged well before reaching this limit.

5.2 Performance metrics

Here is a presentation of the performance metrics that are used for comparison.

Comparing the parameters

To assess the performance of the algorithm regarding the parameters, we use the **Mean Squared Error** (MSE), commonly used in statistics to measure the quality of predictions. It quantifies the average squared difference between the actual values and the values estimated by the model. Mathematically, the MSE is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where:
· n is the number of observations,
· y_i represents the actual value for the i-th observation,
· $\hat{y}_i$ denotes the estimated value for the i-th observation.

The lower the value of the MSE, the better the predictive performance is.

The **standard deviation** (sd) of the prediction errors is also an important metric that we use here to evaluate the accuracy of the model. It measures the dispersion of the prediction errors and thus reflects the variability of the estimation's accuracy.
As for the MSE, smaller values for the standard deviation are preferred.

Comparing the classifications

The **Adjusted Rand Index** (ARI) is a measure used to evaluate the similarity between two clusterings or partitions, which is an adjustment of the Rand Index (RI) to account for the chance grouping of elements, providing a more accurate assessment of clustering quality.

The RI measures the similarity between two clusterings by checking if pairs of points are either in the same group or in different groups in both clusterings. A pair is called a match when both points are simultaneously either in the same group or in different groups in both clusterings. The value of the RI is a ratio, given by the following formula:

$$\text{RI} = \frac{\text{Number of matches}}{\text{Total number of pairs}}$$

The Rand Index ranges from 0 (completely different clusterings) to 1 (identical clusterings).

The ARI is then computed by adjusting the RI to account for the fact that random clusterings would result in some level of agreement just by chance. This makes the ARI more reliable when comparing clusterings with different numbers of clusters, or when comparing clusters of varying sizes. This adjustment is calculated by:

$$\text{ARI} = \frac{\text{RI} - \text{Expected RI}}{1 - \text{Expected RI}}$$

where the Expected RI is the expected value of the Rand Index under a random model, meaning the value one would expect if the two clusterings were completely random. This expected value depends on both the number of clusters and the distribution of elements within those clusters.
The ARI ranges from -1 (completely dissimilar) to 1 (perfect match), with 0 indicating random labeling. It is a metric that is widely used for evaluating clustering algorithms, as it can effectively handle situations where clusters are of different sizes or when the number of clusters differ.

Comparing the number of clusters

To evaluate the ability of the BIC to recover the true number of components in the population, we simply compare the quantity of simulated datasets (among the 100 generated per configuration) that have recovered the right number of components G=2, instead of G=1.

5.3 Results

The results of the simulation study are shown in Table 3, across different sample sizes (n) and different total lengths of sequences. For each configuration, are shown the mean value of the ARI across the 100 simulated datasets, as well as the mean MSE and the standard deviation across all the parameter estimates, as defined in the previous section. All values are rounded to 10^{-4} to ensure more clarity in the table.

In Table 3, π represents the mixture probabilites, α the initial probabilities, P the transitions matrices, and Γ_a and Γ_λ respectively represent the shape and the rate parameters of the gamma distributions.

The results show that the estimations are overall promising. Firstly, the model performs very well in selecting the correct number of mixture components. Notably, the selection is perfect when the sample is of size 250 or 500, or when the sequences lengths exceed 20, where the BIC consistently selects $G = 2$.

Similarly, we see that the classification of the sequences into their respective groups improves rapidly, with the ARI increasing as the lengths of the sequences get longer, and that is in average equal to 1, the highest value it can take, when the sequences are longer than 20 visited states, but that is already very good with shorter sequences.

Table 3: Simulation study results for different sample sizes and sequences lengths

n = 100				
Sequences Lengths		**3 to 10**	**10 to 20**	**20 to 30**
Chosen G		*1* *2* 13 87	*1* *2* 2 98	*1* *2* 0 100
ARI		0.627	0.959	1.000
π	MSE	0.0139	0.0085	0.0029
	sd	0.1184	0.0925	0.0539
α	MSE	0.0157	0.0065	0.0038
	sd	0.1255	0.0807	0.0616
P	MSE	0.0558	0.0113	0.0050
	sd	0.2364	0.1063	0.0710
Γ_a	MSE	6.2439	2.6478	0.2104
	sd	2.2898	1.6143	0.4434
Γ_λ	MSE	1.6000	1.0000	0.3159
	sd	1.1311	0.9544	0.5405

n = 250				
Sequences Lengths		**3 to 10**	**10 to 20**	**20 to 30**
Chosen G		*1* *2* 0 100	*1* *2* 0 100	*1* *2* 0 100
ARI		0.762	0.994	1.000
π	MSE	0.0033	0.0008	0.0010
	sd	0.0576	0.0284	0.0310
α	MSE	0.0103	0.0014	0.0014
	sd	0.1017	0.0376	0.0381
P	MSE	0.0480	0.0004	0.0003
	sd	0.2192	0.0198	0.0164
Γ_a	MSE	1.7927	0.3819	0.1515
	sd	1.0944	0.5549	0.3611
Γ_λ	MSE	1.4898	0.4598	0.2224
	sd	1.0822	0.6115	0.4316

n = 500				
Sequences Lengths		**3 to 10**	**10 to 20**	**20 to 30**
Chosen G		*1* *2* 0 100	*1* *2* 0 100	*1* *2* 0 100
ARI		0.784	0.996	1.000
π	MSE	0.0024	0.0004	0.0006
	sd	0.0490	0.0194	0.0247
α	MSE	0.0120	0.0010	0.0006
	sd	0.1096	0.0318	0.0255
P	MSE	0.0612	0.0017	0.0002
	sd	0.2474	0.0416	0.0134
Γ_a	MSE	1.8599	0.3649	0.1255
	sd	1.0990	0.5335	0.3231
Γ_λ	MSE	1.6592	0.4761	0.1922
	sd	1.1388	0.6188	0.3965

As for the parameters, the estimations are generally very close to their true values, even with small sample sizes and short sequences. The main exception is the gamma parameters, particularly the shape parameter, where estimates deviate more from their true values when both the sample size and the number of visited states are small. However, accuracy improves quickly as either the sample size or sequence length increases. For the other parameters — namely the mixture probabilities, the initial probabilities and the transition matrices — the estimates are already accurate, even when the sample size is small (n=100) with short sequences (from 3 to 10), which is very encouraging.

Without big surprise, Table 3 overall confirms that larger sample sizes and sequence lengths lead to better performance, both in terms of clustering and parameter estimation, as evidenced by the decreased MSE and standard deviation across all parameters.

In practice, we can expect that there may not always be many visited states; indeed, relatively short sequences are not uncommon in sociological trajectories. Therefore, to obtain the most accurate estimates in such cases, it is important to aim at having a sufficient number of individuals in the dataset.

6 Conclusion

In this work, we have explored finite mixtures of semi-Markov chains and their application to the clustering of sociological trajectories, focusing on sequences described by categorical variables.

We have first presented the traditional approaches commonly used nowadays for this matter, and have then defined the theoretical framework of our probabilistic model. We have thereafter presented in detail how one can implement this model, based on Cardot et al.'s work [1], and have detailed the consecutive steps of the algorithm as to how it operates to return results: first through an Expectation-Maximisation algorithm for the parameter estimates, then with the Bayesian Information Criterion to select the optimal number of clusters, and finally using the Maximum A Posteriori classification tool to assign sequences to their corresponding groups. Finally, the evaluation of this probabilistic method on simulated data has shown good performance, and allows us to confidently affirm that this approach, as has been defined in this work, is able to yield promising and reliable results, as soon as certain conditions are verified.

We have been able to demonstrate that this semi-Markov mixture approach presents significant advantages over traditional distance-based methods such as Optimal Matching. Notably, the model provides a more rigorous framework, avoiding the arbitrary cost settings required by Optimal Matching. Moreover, the interpretation of distance-based results relies heavily on visualisation, as we can only assess the various clusters visually and separately. In contrast, the semi-Markov approach allows us to extract interpretable parameters, offering a more insightful understanding of the processes underlying the sociological sequences in the data. Those parameters offer precise insights into the specific behaviours of each cluster, information that is valuable for sociologists studying a set of sequences.

An area, however, where the proposed approach could be improved is in addressing the assumption of time homogeneity and the memoryless nature of Markov chains. Indeed, the resulting transition matrices in the model are assumed to remain constant over the span of the sequences within a cluster. This aspect may reduce the model's applicability in certain social science contexts, where individuals' behaviours are likely to evolve over time.

While the mathematical foundations of the semi-Markov mixture model are more demanding than the traditional approaches, it offers clearer distinctions between population groups and produces thorough analytical results for sociological interpretation. This enhanced interpretability marks a true step forward, suggesting that such models hold great potential for future studies in the field of sociology, as a robust and rigorous alternative to the commonly used methods.

References

[1] Hervé Cardot, Guillaume Lecuelle, Pascal Schlich, and Michel Visalli. Estimating finite mixtures of semi-markov chains: an application to the segmentation of temporal sensory data. 2018.

[2] Matthias Studer and Gilbert Ritschard. What matters in differences between life trajectories: a comparative review of sequence dissimilarity measures. 2015.

[3] Tim F. Liao, Danilo Bolano, Christian Brzinsky-Fay, Benjamin Cornwell, Anette Eva Fasang, Satu Helske, Raffaella Piccarreta, Marcel Raab, Gilbert Ritschard, Emanuela Struffolino, and Matthias Studer. Sequence analysis: Its past, present, and future. 2022.