République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

Université de Saida - Dr Moulay Tahar.
Faculté des Sciences.
Département de Mathématiques.

Mémoire présenté en vue de l'obtention du diplôme de

Master Académique

Filière : MATHEMATIQUES
Spécialité: Analyse Stochastique Statistique des Processus et Applications

par

Melle.Elias **A**bla[1]

Sous la direction de

Dr. F. Mokhtari

Thème:

HIDDEN SEMI MARKOV MODEL AND ESTIMATION

Soutenue le 16/06/2022 devant le jury composé de

Dr.A. Bouaka	Université de Saïda Dr. Moulay Tahar	Président
Dr.F. Mokhtari	Université de Saïda Dr. Moulay Tahar	Encadreur
Dr.R. Hazeb	Université de Saïda Dr. Moulay Tahar	Examinateur

Année univ.: 2021/2022

[1] e-mail: ablaboch671@gmail.com

⋇ *Dedication* ⋇

All praise to ***ALLAH****, all glory to him*
all praise to our prophet ***Muhammad***
thank you for making this day happened, thank you for making us reach the end of our work and efforts.
As well as everything that I do. I would be honor to dedicate this humble work
to:
My parents*, the two person that gave me the tools and values necessary to be where I am standing today.*
To all ***the family members****, thank you one by one for the support and the warmth you give me.*
To my sister ***AMEL*** *and my little brother* ***BEN OUTMAN****. God bless you and protect you for me.*
To all the friends and the people who loved me for who I am.
For all those if my pen forgets them, my heart will not forgotten them.
For all my sisters and brothers in Islam.

⋇ *Acknowledgments* ⋇

In the Name of Allah, that we start everything by his name, all praise be to Allah and prayers and peace be upon Mohammed His servant and messenger.
First and foremost, I must acknowledge my limitless thanks to Allah, the one and only one I must write his name whenever I hold this pen, because of his blessings and his help and his guidance.
I would like also to express my wholehearted thanks to my supervisor Madam **Dr.F.Mokhtari**, who worked hard with me from the beginning till the end, and I can not deny her work and efforts in this work. I greatly appreciate her support and her encouragement and patience throughout this master memory.
I wish to thank my master memory committee members: **Dr.A.Bouekka**, **Dr.R.Hazeb** for generously offering their time, support, guidance and good will throughout the preparation and review of this document.
I wish to thank all the family members for everything they done to me, my love for you all can not be expressed and so the gratitude that I owe you is non expressible.
My love goes of both of my sisters **AMEL** and **NESRINE**, and my brother **NACER**, without forgetting my best friend **SAMIHA**, I wish you all the happiness and the good health.
My love goes to all the spoilers kids in the house **ABED ALLAH, NOUR EL DINE, ABED EL RAHMANE, FARAH, NADIR, YASSINE, INESE, SONIA, RIADH, ISLEM, IBRAHIM, MERIEM**. God protect them all.
Last but not least, I wish all the good for all the people in this life, at hope we see each other in a good place, where is the peace and the love shared with no limits .

Contents

Dedication **2**

Acknowledgments **3**

Notations **6**

Introduction **9**

1 Markov and semi Markov model **12**

1.1 Introduction and preliminaries 12

1.1.1 Preliminaries 14

1.2 Discrete-time Markov model 16

1.2.1 State classification 19

1.3 Discrete-time semi Markov model 20

1.3.1 Markov renewal chains 20

1.3.2 Definition of semi-Markov chains and sojourn time distributions 21

1.4 Elements of statistical estimation 24

1.4.1 Empirical estimators 24

1.5 Asymptotic properties of the estimators 27

1.5.1 Strong consistency 27

1.5.2 Asymptotic normality 29

2 Hidden Markov and semi Markov models **34**

2.1 Introduction 34

2.2 Hidden Markov model 35

2.2.1 Three basic problems of HMM and their solutions 41

2.3 Hidden semi-Markov model 48

2.3.1 General structure of HSMM 48

2.3.2 Estimation of a Hidden Semi-Markov Model 53

2.3.3 Consistency of Maximum-Likelihood Estimator 54

2.3.4 Asymptotic Normality of Maximum-Likelihood Estimator . . . 56

3 R packages for analyzing SMM, HMM, HSMM 58

3.1 Introduction 58

3.2 Package SMM 58

3.2.1 Simulation of semi-Markov models 59

3.2.2 Estimation of semi-Markov models 62

3.3 Package HMM 65

3.4 Package hsmm 72

3.4.1 Simulation of observation and state sequences 72

3.4.2 Maximum likelihood estimation of the model parameters . . . 74

3.4.3 Inference on the underlying state sequence 78

Bibliography 79

Notations

$\mathbb{N}$	**Set of positive natural numbers**.
$(\Omega, \mathcal{F}, \mathbb{P})$	**Probability space**.
$\mathbb{E}$	**Expectation with respect to** $\mathbb{P}$.
$\mathrm{E} = \{1, \dots, s\}$	**Finite state space.**
$\mathcal{M}_{\mathrm{E}}$	**Set of real matrix on** $\mathrm{E} \times \mathrm{E}$.
$\mathcal{M}_{\mathrm{E}}(\mathbb{N})$	**Matrix-valued functions defined on** $\mathbb{N}$**,** **with values in** $\mathcal{M}_{\mathrm{E}}$.
$Z := (Z_k)_{k \in \mathbb{N}}$	**Semi-Markov chain (SMC)**.
$(J, S) := (J_n, S_n)_{n \in \mathbb{N}}$	**Markov renewal chain (MRC)**.
$J := (J_n)_{n \in \mathbb{N}}$	**Visited states, embedded Markov chain (EMC)**.
$S := (S_n)_{n \in \mathbb{N}}$	**Jump times**.
$L := (L_n)_{n \in \mathbb{N}}$	**Sojourn times.**
M	**Fixed censoring time**.
$N(M)$	**Number of jumps of Z in the time interval** $[1, M]$.
$N_i(M)$	**Number of visits to state i of the EMC, up to time M**.
$N_{ij}(M)$	**Number of transitions from state i to state j of the EMC, up to time M**.
$N_{ij}(k, M)$	**Number of transitions from state i to state j of the EMC, up to time M, with sojourn time in state i equal to k**.
$\mathbf{p} := (p_{ij})_{i,j \in \mathrm{E}}$	**Transition matrix of the EMC J**.
$\mathbf{q} := (q_{ij}(k))_{i,j \in \mathrm{E}, k \in \mathbb{N}}$	**Semi-Markov kernel**.

$\mathbf{Q} := (Q_{ij}(k))_{i,j\in\mathrm{E},k\in\mathbb{N}}$	Cumulated semi-Markov kernel.
$\mathbf{f} := (f_{ij}(k))_{i,j\in\mathrm{E},k\in\mathbb{N}}$	Conditional sojourn time distribution in state i, before visiting state j.
$\mathbf{F} := (F_{ij}(k))_{i,j\in\mathrm{E},k\in\mathbb{N}}$	Conditional cumulative sojourn time distribution in state i, before visiting state j.
$\mathbf{h} := (h_i(k))_{i\in\mathrm{E},k\in\mathbb{N}}$	Sojourn time distribution in state i.
$\mathbf{H} := (H_i(k))_{i\in\mathrm{E},k\in\mathbb{N}}$	Cumulative distribution of sojourn time in state i.
$\overline{\mathbf{H}} := (\overline{H}_i(k))_{i\in\mathrm{E},k\in\mathbb{N}}$	Survival function in state i.
μ_{ij}	Mean first passage time from state i to state j, for semi Markov process Z.
μ_{ij}^*	Mean first passage time from state i to state j, for embedded Markov chain J.
$\nu = (\nu(j)_{j\in\mathrm{E}})$	Stationary distribution of the EMC J.
$\alpha = (\alpha_i)_{i\in\mathrm{E}}$	Initial distribution of semi-Markov process Z.
$A * B$	Discrete-time matrix convolution product of A, B.
$A^{(n)}$	n-fold convolution of $A \in \mathcal{M}_{\mathrm{E}}(\mathbb{N})$.
$X := (X_n)_{n\in\mathbb{N}}$	Unobserved MC (hidden).
$Y := (Y_n)_{n\in\mathbb{N}}$	Observable process.
$b_j(k)_{1\leq i\leq N,1\leq k\leq M}$	The emission probability between state s_j and observation r_k.
$\mathbf{A}$	Transition probability matrix in HMM or HSMM.
$\mathbf{B}$	Emission probability matrix in HMM or HSMM.
λ	The parameters of HMM or HSMM.
P_λ	The probability given a model λ.
$\gamma_n(i_n)$	Forward variable.
$\beta_n(i_n)$	Backward variable.
$\rho_n(i_n)$	The highest probability along a path (viterbi variable).
$\phi_n(i_n)$	The argument which maximize $\rho_n(i_n)$.
$\xi_n(i_n, i_{n+1})$	The probability of being in state s_{i_n} at time n and state $s_{i_{n+1}}$ at time $n+1$.
$\omega_n(i_n)$	The probability of being in state s_{i_n} at time n.
$U := (U_n)_{n\in\mathbb{N}}$	Backward-recurrence times of the SMC Z.
$(Z, Y) := (Z_n, Y_n)_{n\in\mathbb{N}}$	Hidden semi-Markov chain (Z hidden component).
$\pi := (\pi_j)_{j\in E}$	Limit distribution of semi-Markov chain Z.

$\mathrm{A} = \{1, \ldots, d\}$	**Finite state space of chain Y.**
$\mathbf{R} := (R_{i,a})_{i \in E, a \in A}$	**Conditional distribution of Y, given** $\{Z_n = i\}$**, for** (Z, Y) **hidden chain SM1-M0.**
$\xrightarrow{a.s}$	**Almost sure convergence (strong consistency).**
$\xrightarrow{P}$	**Convergence in probability.**
$\xrightarrow{\mathcal{D}}$	**Convergence in distribution.**
δ_{ij}	**Symbole of Kronecker.**
$\mathbb{1}_A$	**Indicatrice function of A.**
$\mathcal{N}(0, \sigma^2)$	**Standard normal random variable** **(mean** $\mu = 0$ **,variance** σ^2**).**
$DTMP$	**Discrete-time Markov process.**
MC	**Markov chain.**
SMC	**Semi-Markov Chain.**
RC	**Renewal Chain.**
EMC	**Embedded Markov Chain.**
MLE	**Maximum-Likelihood Estimator.**
$SLLN$	**Strong Law of Large Numbers.**
CLT	**Central Limit Theorem.**
$r.v$	**random variable.**
RP	**Renewal Process.**
HMM	**Hidden Markov Model.**
$HSMM$	**Hidden Semi-Markov Model.**

Introduction

In various sectors, such as nuclear and power plants, communication networks, biological systems, software reliability, DNA analysis, insurance and finance, earthquake modeling, etc., the systems are becoming more and more complex. In recent years, there has been growing interest in evaluating the performance of systems. The evolution of a system is modeled by a stochastic process. Among the models which are widely used as a standard tool to describe the evolution of a system, we have the Markov and the semi-Markov models. Markov defined a way to represent real-world stochastic systems and processes that encode dependencies and reach a steady-state over time.

One of the reasons for applying Markov process theory in various fields is that the Markovian hypothesis is very intuitive: if we know the past and present of a system, then the future development of the system is only determined by its present state. So, the history of the system does not play a role in its future development. We also call this the memoryless property. However, the Markov property has its limitations. It enforces restrictions on the distribution of the sojourn time in a state, which is exponentially distribution (continuous case) or geometrically distribution (discrete case). This is a disadvantage when we apply Markov processes in real-life applications.

Therefore, we can introduce the semi-Markov process. This process allows us to have arbitrary distributed sojourn time in any state and still provides the Markov property, but in a more flexible way. The memoryless property does not act on the calendar time in this case, but on the sojourn time in the state.

The semi Markov processes are often used during the functioning of the system for which the semi Markov model is built, but it is not always possible to get all the information contained in the status codes when changing its states, instead only we can get is the signal (symbols) in which block of system elements the state changed (failure, renewal, etc.,). In this case, the states of the semi Markov model can be considered hidden (unobservable), and so we called it hidden semi Markov model (HSMM).

A hidden semi Markov model (HSMM) is traditionally defined by allowing the underlying process to be a semi Markov chain. Each state has a variable duration, which is associated with the number of observations being produced while in the state. This makes it suitable for use in a wider range of applications.

The first approach to hidden semi Markov model was proposed by Ferguson [15], which is partially included in the survey paper by Rabiner [26]. This approach is called the explicit duration HMM in contrast to the implicit duration of the HMM. It assumes that the state duration is generally distributed depending on the current state of the underlying semi Markov process. It also assumes the conditional independence of outputs.

As Ferguson [15] pointed out, an HSMM can be realized in the HMM framework in which both the state and its sojourn time since entering the state are taken as a complex HMM state. This idea was exploited in 1991 by a 2 vector HMM [17] and a duration dependent state transition model [29]. Since then, similar approaches were proposed in many applications. These approaches, however, have the common problem of computational complexity in some applications. A more efficient algorithm was proposed in 2003 by Yu and Kobayashi [30], in which the forward-backward variables are defined using the notion of a state together with its remaining sojourn (or residual life) time. This makes the algorithm practical in many applications.

The HSMM has been successfully applied in many areas. The most successful application is in speech recognition [15]. The first application of HSMM in this area was made by Ferguson. Since then, there have been more than one hundred such papers published in the literature. It is the application of HSMM in speech recognition that enriches the theory of HSMM and develops many algorithms for HSMM. Since the beginning of 1990's, the HSMM started being applied in many other areas such as printed text recognition [1] or handwritten word recognition [19], recognition of human genes in DNA [18], language identification [23], etc.

This master memory falls into three chapters.

In chapter 1, we combined the two models Markov and semi Markov model. We give some background and some basic concepts, properties, and theorems on homogeneous Markov chains with a discrete set of states (our work concern only with the case of finite set state). For semi Markov chain, We give its basic probabilistic properties and we present their empirical estimators for the main characteristics accompanied with their asymptotic properties (the strong consistency, the asymptotic normality).

In chapter 2, we consider the Markov chain and the semi Markov chain as an unobserved processes in discrete time, and their observation sequence produced by the hidden states, to describe another model called the hidden Markov model and the hidden semi Markov model, then we continue by giving the definition of this two models and we present the basic problems of the hidden Markov, then we give the algorithms that can solve them consists of the Forward and Backward algorithm and the viterbi algorithm and the Baum-Welch algorithm (EM algorithm), and we finish by giving the maximum likelihood estimation of hidden semi Markov model, and their asymptotic properties (the strong consistency, the asymptotic normality).

In chapter 3, we present the R packages **SMM, HMM, hsmm** used for the simulation and non parametric estimation of discrete-time respectively of the semi Markov models, hidden Markov, and Hidden semi-Markov models, and we give a detailed description of each packages with some examples.

Chapter 1

Markov and semi Markov model

1.1 Introduction and preliminaries

Andrei Markov didn't agree with Pavel Nekrasov, when he said independence between variables was necessary for the **Weak Law of Large Numbers** to be applied. The Weak Law of Large Numbers states something like this:

⊸***When you collect independent samples, as the number of samples gets bigger, the mean of those samples converges to the true mean of the population.***⊸

But Markov believed independence was not a necessary condition for the mean to converge. So he set out to define how the average of the outcomes from a process involving dependent random variables could converge over time. Thanks to this intellectual disagreement, Markov created a way to describe how random, also called stochastic systems or processes evolve over time. The system is modeled as a sequence of states and, as time goes by, it moves in between states with a specific probability. Since the states are connected, they form a chain. Following the academic tradition of naming discoveries and new methods after the people that developed or discovered them, this way of modeling the world is called a Markov Chain. What's particular about Markov chains is that, as you move along the chain, the state where you are at any given time matters. The transitions between states are conditioned, or dependent, on the state you are in before the transition occurs, that's what is named as the Markov property. Putting all of these characteristics together, Markov was able to prove that, as long as you can reach all states in the chain, the probability of moving to a particular state will converge to a single steady value in the long run.

The Markov property imposes restrictions on the distribution of the sojourn time in the state, and for that we define the semi Markov model which allows us to have arbitrary distributed sojourn time in any state. The semi-Markov models are more general than Markov models because they are not limited by the Markov assumption, they were introduced by Levy [21] and Smith [27] in 1950s and are applied in queuing theory and reliability theory.

Definition 1.1.1. ***(Probability measures)***

It all begins with a ***probability measure*** $\mathbb{P}$. *You should think of a probability measure* $\mathbb{P}$*, on a set* Ω *as a function assigning a number* $\mathbb{P}(A) \in [0,1]$ *to subsets* $A \subset \Omega$. *If you are familiar with measure theory you may correctly insist that a probability measure only assigns a probability to subsets* $\subset \mathcal{F}$ *in a* σ*-algebra* $\mathcal{F}$ *on* Ω[1] *but this point of view is not crucial for the story to come. Subsets of* Ω *are referred to as* ***events****.*

By definition a probability measure must have total mass equal to one and it must be additive over countable classes of disjoint sets, i.e.

$$\mathbb{P}(\bigcup_{n=1}^{\infty} A_n) = \sum_{n=1}^{\infty} \mathbb{P}(A_n)$$

provided that $A_i \bigcap A_j = \emptyset, i \neq j$.

For two events A,B *with* $\mathbb{P}(B) > 0$ *we define the elementary* ***conditional probability*** *of* A *given* B *(notation:* $\mathbb{P}(A \mid B)$*) as*

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \bigcap B)}{\mathbb{P}(B)}$$

.

Definition 1.1.2. ***(Random variables)***

When we refer to a random experiment we want to emphasize that we are in a situation where we are unable to predict the outcome with certainty. There might be several reasons that we do not know the exact result of an experiment: the outcome may be affected by circumstances that we are unable to control or we may simply not have complete information allowing us to determine the result of the experiment.

[1] A σ-algebra on Ω is a class $\mathcal{F}$ of subsets of Ω such that,

- $\Omega \in \mathcal{F}$
- $A^c \in \mathcal{F}$ if $A \in \mathcal{F}$
- $\bigcup_{n=1}^{\infty} A_n \in \mathcal{F}$ if $A_1, A_2, \ldots, \in \mathcal{F}$

*The concept of a **random variable** or **stochastic variable** is used for a mathematical model of a random experiment. Formally, a random variable, "X", is a function and we reflect the randomness by saying that the argument $\omega \in \Omega$ of the function is chosen according to some probability distribution, $\mathbb{P}$. The outcome of the experiment is denoted by $X(\omega)$. Two different ω's will potentially give different results of the experiment reflecting the non-deterministic nature of the experiment.*

A random variable may be defined (more formally) as a measurable map

$$X : (\Omega, \mathcal{F}) \longrightarrow (E, \mathcal{G})$$

where $\mathcal{F}, \mathcal{G}$ are classes of subset satisfying the conditions of a σ-algebra. For a subset $A \subset E$ (with $A \subset G$) the probability that the random experiment gives a value in the set A is computed as

$$\mathbb{P}(X \in A) \equiv \mathbb{P}(\{\omega \in \Omega \mid X(\omega) \in A\})$$

. This defines a probability on E which we will call the distribution of random variable X.

Definition 1.1.3. ***(Stochastic processes)***

A stochastic process is a family of random variables indexed by a set I, $\{X(t), t \in I\}$ defined on $(\Omega, \mathcal{F}, \mathbb{P})$ with values in E. For every $t \in I$, $X(t)$ is a random variable $X(t) : \Omega \to \mathrm{E}$, whose value for the outcome $\omega \in \Omega$ is noted $X(t, \omega)$. If instead of t we fix an $\omega \in \Omega$, we obtain the function $X(., \omega) : I \to \mathrm{E}$ which is called a trajectory or a path-function or a sample function of the process.

The set E is called the state space of the stochastic process $X = (X(t), t \in I)$. The stochastic process may be denoted by X_t instead of $X(t)$ (respectively, X_n if $I = \mathbb{N}$ (discrete time random process)).

Consider a finite set $\mathrm{E} = \{1, \ldots, s\}$. We denote by $\mathcal{M}_E$ the set of real matrices on $\mathrm{E} \times \mathrm{E}$ and by $\mathcal{M}_E(\mathbb{N})$ the set of matrix valued functions defined on $\mathbb{N}$, with values in $\mathcal{M}_E$.

1.1.1 Preliminaries

In this sequel we introduce some notations and theorems which will be useful later.

Theorem 1.1.1. ***(Strong Law of Large Numbers)****[22]*

Let $(X_1, X_2, \ldots)$ is an infinite sequence of i.i.d. Lebesgue integrable random variables with expected value $\mathbb{E}[X_1] = \mathbb{E}[X_2] = \ldots$, then we have

$$\frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow[n\to\infty]{a.s} \mathbb{E}[X_1].$$

Theorem 1.1.2. ***(Glivenko-Cantelli theorem)*** *[11]*

Let $F_n(x) = \frac{1}{n}\sum\limits_{k=1}^{n} \mathbf{1}_{\{X_k \leq x\}}$ *be the empirical distribution function of the i.i.d. random sample* $X_1, \ldots, X_n$*. Denote by* F *the common distribution function of* X_i *,* $i = 1, \ldots, n$*. Thus*

$$\sup_{x\in\mathbb{R}} |F_n(x) - F(x)| \xrightarrow[n\to\infty]{a.s} 0.$$

Theorem 1.1.3. *[16]*

Let $(Y_n)_{n\in\mathbb{N}}$ *be a sequence of random variables and* $(N_n)_{n\in\mathbb{N}}$ *a positive integer-valued stochastic process. Suppose that*

$$Y_n \xrightarrow[n\to\infty]{a.s} Y \text{ and } N_n \xrightarrow[n\to\infty]{a.s} \infty.$$

Then,

$$Y_{N_n} \xrightarrow[n\to\infty]{a.s} Y.$$

Definition 1.1.4. ***(Martingale)***

Let $\mathbf{F} = (\mathcal{F}_n, n \geq 0)$ *be a family of sub-*σ*-algebras of* $\mathcal{F}$ *such that* $\mathcal{F}_n \subset \mathcal{F}_m$*, when* $n < m$*. We say that* $\mathbf{F}$ *is a filtration of* $\mathcal{F}$*. A real-valued* $\mathbf{F}$*-adapted stochastic process* X_n *is (*$\mathcal{F}_n$*-measurable for* $n \geq 0$*) called martingale with respect to a filtration* $\mathbf{F}$ *if, for every* $n = 0, 1, \ldots$

1. $\mathbb{E}|X_n| < \infty$ *; and*

2. $\mathbb{E}[X_{n+1}|\mathcal{F}_n] = X_n$ *(a.s).*

Theorem 1.1.4. ***(CLT for martingales)****[9]*

Let $(X_n)_{n\in\mathbb{N}^\star}$ *be a martingale with respect to the filtration* $\mathcal{F} = (\mathcal{F}_n)_{n\in\mathbb{N}}$ *and define the process* $Y_n = X_n - X_{n-1}$*,* $n \in \mathbb{N}^\star$ *(with* $Y_1 := X_1$*), called a difference martingale. If*

1. $\frac{1}{n}\sum\limits_{k=1}^{n} \mathbb{E}[Y_k^2|\mathcal{F}_{k-1}] \xrightarrow[n\to\infty]{P} \sigma^2 > 0;$

2. $\frac{1}{n}\sum\limits_{k=1}^{n} \mathbb{E}[Y_k^2\mathbf{1}_{\{|Y_k|>\epsilon\sqrt{n}\}}] \xrightarrow[n\to\infty]{} 0,$ *For all* $\epsilon > 0$*,*

then

$$\frac{X_n}{n} \xrightarrow[n\to\infty]{a.s} 0,$$

and

$$\frac{1}{\sqrt{n}} X_n = \frac{1}{\sqrt{n}} \sum_{k=1}^{n} Y_k \xrightarrow[n\to\infty]{\mathcal{D}} \mathcal{N}(0,\sigma^2).$$

Theorem 1.1.5. ***(Anscombe's theorem)[10]***

Let $(Y_n)_{n\in\mathbb{N}}$ be a sequence of random variables and $(N_n)_{n\in\mathbb{N}}$ a positive integer-valued stochastic process. Suppose that

$$\frac{1}{\sqrt{n}} \sum_{m=1}^{n} Y_m \xrightarrow[n\to\infty]{\mathcal{D}} \mathcal{N}(0,\sigma^2) \quad \textit{and} \quad N_n/n \xrightarrow[n\to\infty]{P} \theta,$$

where θ is a constant, $0 < \theta < \infty$. Then,

$$\frac{1}{\sqrt{N_n}} \sum_{m=1}^{N_n} Y_m \xrightarrow[n\to\infty]{\mathcal{D}} \mathcal{N}(0,\sigma^2).$$

1.2 Discrete-time Markov model

In this section we are only going to deal with a very simple class of mathematical models for random events namely the class of Markov chains on a finite state space (The state space is the set of possible values for the observations).

Definition 1.2.1. ***(Discrete-time Markov chain)***

Let $(J_n)_{n\geq 0}$ be a stochastic process defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with values in a finite set $\mathrm{E} = \{1, 2, \dots, s\}$.

1. *A stochastic process $(J_n)_{n\geq 0}$ is called discrete time Markov process or Markov chain with state space E if, for any $n \in \mathbb{N}$ and any state sequence $i_0, i_1, \dots, i_{n-1}, i, j$ $\in \mathrm{E}$,*

$$\mathbb{P}(\underbrace{J_{n+1} = j}_{Future} \,|\, \underbrace{J_0 = i_0, J_1 = i_1, \dots, J_{n-1} = i_{n-1}, J_n = i}_{Past\,and\,present}) = \mathbb{P}(\underbrace{J_{n+1} = j}_{Future} | \underbrace{J_n = i}_{Present}). \tag{1.1}$$

the equality (1.1) is called ***Markov property****.*

2. *If, additionally, the probability $\mathbb{P}(J_{n+1} = j | J_n = i)$ does not depend on n, $(J_n)_{n\geq 0}$ is said to be homogeneous with respect to time.*

Definition 1.2.2. ***(Transition Matrix)***

The function $(i, j) \to p_{ij} := \mathbb{P}(J_{n+1} = j | J_n = i)$ *is called transition function of the chain. For any* $i, j \in \mathrm{E}$ *and* $n \geq 0$, *the transition function satisfies the following properties :*

1. $p_{ij} \geq 0$, *for any* $i, j \in \mathrm{E}$,
2. $\sum_{j \in \mathrm{E}} p_{ij} = 1$, *for any* $i \in \mathrm{E}$,
3. $\sum_{k \in \mathrm{E}} p_{ik} p_{kj} = \mathbb{P}(J_{n+2} = j | J_n = i) = p_{ij}^{(2)}$,
4. $\sum_{k \in \mathrm{E}} p_{ik}^{(n)} p_{kj}^{(m)} = p_{ij}^{(n+m)}$, *for any* $i, j \in \mathrm{E}$ *and* $n, m \geq 0$.

as we are consider only with finite state space Markov chains, we can represent the transition function by a squared matrix (transition matrix $\mathbf{p} \in \mathcal{M}_{\mathrm{E}}$*):*

$$\mathbf{p} = (p_{ij})_{i,j \in \mathrm{E}} = \begin{pmatrix} p_{11} & \cdots & p_{1s} \\ \vdots & & \vdots \\ p_{s1} & \cdots & p_{ss} \end{pmatrix}$$

In order to define a Markov chain $(J_n)_{n \geq 0}$ we need:

1. transition function (matrix) $\mathbf{p} = (p_{ij})_{i,j \in \mathrm{E}}$.
2. $\alpha = (\alpha_1, \ldots, \alpha_s)$, the initial distribution of the chain, that is the distribution of J_0, $\alpha_i = \mathbb{P}(J_0 = i)$ for any state $i \in \mathrm{E}$.

Proposition 1.2.1. *Let* $(J_n)_{n \in \mathbb{N}}$ *be a Markov chain of transition function* $\mathbf{p}$ *and initial distribution* α. *For any* $n \geq 1$, $k \geq 0$, *and any states* $i_0, i_1, \ldots, i_n, i, j \in \mathrm{E}$, *we have:*

1.

$$\mathbb{P}(J_{k+1} = i_1, \ldots, J_{k+n-1} = i_{n-1}, J_{k+n} = i_n \mid J_k = i_0) = p_{i_0 i_1} \ldots p_{i_{n-1} i_n} \qquad (1.2)$$

2.

$$\mathbb{P}(J_0 = i_0, J_1 = i_1, \ldots, J_{n-1} = i_{n-1}, J_n = i_n) = \alpha_{i_0} p_{i_0 i_1} \ldots p_{i_{n-1} i_n}. \qquad (1.3)$$

Proof.

1. *We put* $A_j = \{J_{k+j} = i_j\}$ *for all* $j \in 0, \dots, n$ *we have:*

   $$\mathbb{P}(A_n \cap \dots \cap A_1 \mid A_0) = \mathbb{P}(A_n \mid A_{n-1} \cap \dots \cap A_0)\mathbb{P}(A_{n-1} \mid A_{n-2} \cap \dots \cap A_0) \dots \mathbb{P}(A_1 \mid A_0).$$

   Since $(J_n)_n$ *is an homogenous Markov chain, then* $\mathbb{P}(A_j \mid A_{j-1} \cap \dots \cap A_0) = p_{i_{j-1},i_j} = \mathbf{p}(i_{j-1}, i_j)$ *for all* $j \in 1, \dots, n$*, which proves the proposition.*

2. *With the use of the above result and putting k=0:*

   $$\begin{aligned} \mathbb{P}(J_n = i_n, J_{n-1} = i_{n-1}, \dots, J_0 = i_0) &= \mathbb{P}(J_n = i_n, J_{n-1} = i_{n-1}, \dots, J_1 = i_1 | J_0 = i_0) \\ &\times \mathbb{P}(J_0 = i_0) \\ &= \mathbb{P}(J_0 = i_0)\mathbf{p}(i_0, i_1) \dots \mathbf{p}(i_{n-1}, i_n). \end{aligned}$$

   Suppose that for all $n \in \mathbb{N}$ *and for all* $i_0, \dots, i_n \in \mathrm{E}$,

   $$\mathbb{P}(J_n = i_n, J_{n-1} = i_{n-1}, \dots, J_0 = i_0) > 0,$$

   $$\begin{aligned} \mathbb{P}(J_{n+1} = i_{n+1} | J_n = i_n, \dots, J_0 = i_0) &= \frac{\mathbb{P}(J_{n+1} = i_{n+1}, J_n = i_n, \dots, J_0 = i_0)}{\mathbb{P}(J_n = i_n, J_{n-1} = i_{n-1}, \dots, J_0 = i_0)} \\ &= \mathbf{p}(i_n, i_{n+1}). \end{aligned}$$

Proposition 1.2.2. *[4]*

Let $(J_n)_{n \geq 0}$ *be a Markov chain of transition matrix* $\mathbf{p}$.

1. *The sojourn time of the chain in a state* $i \in \mathrm{E}$ *is a geometric random variable on* $\mathbb{N}^*$ *of parameter* $1 - p_{ii}$.

2. *The probability that the chain enters state* j *when it leaves state* i *is* $\frac{p_{ij}}{1-p_{ii}}$ *(for* $p_{ii} \neq 1$*), which means that state i is non absorbing.*

Definition 1.2.3. ***(Stationary distribution)***

A probability distribution ν *on* E *is said to be stationary or invariant for the Markov chain* $(J_n)_{n \geq 0}$ *if, for any* $j \in \mathrm{E}$

$$\sum_{j \in \mathrm{E}} \nu(i) p_{ij} = \nu(j),$$

or, in matrix form,

$$\nu \mathbf{p} = \nu,$$

where $\nu = (\nu(1), \dots, \nu(s))$ *is a row vector.*

1.2.1 State classification

We present in this section some classical ways to characterize a state or an entire Markov chain.

Definition 1.2.4. *(**Accessible state**) We say that state j is accessible from state i, written as $i \to j$ if $p_{ij}^{(n)} > 0$. We assume that every state is accessible from itself since $p_{ii}^{(0)} = 1$.*

Definition 1.2.5. *(**Communicate state**) Two states i and j are said to communicate, written as $i \leftrightarrow j$ if they are accessible from each other. In other words,*

$$i \leftrightarrow j \quad \textit{means} \quad i \to j \quad \textit{and} \quad j \to i.$$

Definition 1.2.6. *(**Irreducible Markov chain**) A Markov chain is said to be irreducible if all states communicate with each other.*

Definition 1.2.7. *(**Recurrent state**) A state is said to be recurrent if, any time that we leave that state, we will return to that state in the future with probability one. On the other hand, if the probability of returning is less than one, the state is called transient. Here, we provide a formal definition:*

For any state i, we define

$$G_{ii} = \mathbb{P}(J_n = i, \textit{for some} \quad n \geq 1 | J_0 = i).$$

State i is recurrent if $G_{ii} = 1$, and it is transient if $G_{ii} < 1$.

Definition 1.2.8. *(**Periodic, aperiodic state**) A state $i \in \mathrm{E}$ is said to be periodic of period $d > 1$, or d-periodic, if d is equal to the greatest common divisor of all n such that $\mathbb{P}(J_{n+1} = i | J_1 = i) > 0$. If $d = 1$, then the state i is said to be aperiodic.*

Definition 1.2.9. *(**Ergodic state**) An aperiodic recurrent state is called ergodic. An irreducible Markov chain with one state ergodic (and then all states ergodic) is called ergodic.*

Proposition 1.2.3. *[4]**(Ergodic theorem for Markov chains).***

For an ergodic Markov chain we have

$$p_{ij}^{n} \underset{n \to \infty}{\longrightarrow} \nu(j),$$

for any $i, j \in \mathrm{E}$.

1.3 Discrete-time semi Markov model

A semi-Markov chain can be analyzed through the so-called embedded renewal chains. That means that by taking into account only some particular aspects of the evolution of a SMC (successive visits of a specific state, for instance), we obtain a renewal chain. Due to this property, results on RCs will be of great help when investigating the behavior of SMCs. In order to define SMM we need to define in first:

Definition 1.3.1. ***(Discrete-Time Renewal Processes)*** *Roughly speaking, a renewal process (RP) $(S_n)_{n\in\mathbb{N}}$ represents the successive instants when a specific (fixed but random) event occurs. The term renewal comes from the assumption that when this event occurs, the process starts anew (this is a regeneration point of time). For this reason, that specific event will be called a renewal and $(S_n)_{n\in\mathbb{N}}$ will be called a renewal process. Since we will be concerned only with discrete-time renewal processes, we will generally use the term renewal chain (RC). For a renewal process we have:*

$$S_n = X_0 + X_1 + \ldots + X_n,$$

if for a certain $n \in \mathbb{N}$ we have $X_n = \infty$, then $S_n = S_{n+1} = \ldots = \infty$. Note that we have $S_0 \leq S_1 \leq \ldots$, where equality holds only for infinite S_n. The sequence $(X_n)_{n\in\mathbb{N}^*}$ is called a waiting time sequence and X_n is the nth waiting time. The sequence $(S_n)_{n\in\mathbb{N}}$ is called an arrival time sequence and S_n is the nth arrival time. Note that the fundamental fact that the chain starts anew each time a renewal occurs means that $(X_n)_{n\in\mathbb{N}^*}$ is a sequence of i.i.d. random variables (in the simplest case, we suppose $X_0 = S_0 = 0$).

1.3.1 Markov renewal chains

Let us consider:

- E the state space. We suppose E to be finite, with $\mid \mathrm{E} \mid= s$.
- The stochastic process $J = (J_n)_{n\geq 0}$ with state space E for the system state at the n^{th} jump.
- The stochastic process $S = (S_n)_{n\geq 0}$ with state space $\mathbb{N}$ for the n^{th} jump. We suppose $S_0 = 0$ and $0 < S_1 < S_2 < \ldots < S_n < S_{n+1} < \ldots$
- The stochastic process $L = (L_n)_{n\geq 0}$ with state space $\mathbb{N}^\star$ for the sojourn time L_n in state J_{n-1} before the n^{th} jump. Thus, $L_n = S_n - S_{n-1}$, for all $n \in \mathbb{N}^\star$.

Definition 1.3.2. ***(Discrete-time semi-Markov kernel)***

A matrix-valued function $\mathbf{q} = (q_{ij}(k)) \in \mathcal{M}_E(\mathbb{N})$ *is said to be a discrete time semi-Markov kernel if it satisfies the following three properties:*

- $0 \leq q_{ij}(k), i, j \in \mathrm{E}, k \in \mathbb{N}$,
- $q_{ij}(0) = 0, i, j \in \mathrm{E}$,
- $\displaystyle\sum_{k=0}^{\infty} \sum_{j \in \mathrm{E}} q_{ij}(k) = 1, i \in \mathrm{E}$.

Definition 1.3.3. ***(Markov renewal chain)***

The stochastic process $(J, S) = (J_n, S_n)_{n \in \mathbb{N}}$ *is said to be a Markov renewal chain (MRC) if for all* $n \in \mathbb{N}$*, for all* $i, j \in \mathrm{E}$ *and for all* $k \in \mathbb{N}$ *it almost surely satisfies:*

$$\mathbb{P}(J_{n+1} = j, S_{n+1} - S_n = k | J_0, \ldots, J_n; S_0, \ldots, S_n) = \mathbb{P}(J_{n+1} = j, S_{n+1} - S_n = k | J_n). \tag{1.4}$$

Moreover, if equation (1.4) is independent of n, (J, S) is said to be homogeneous, with discrete time semi-Markov kernel $\mathbf{q} = (q_{ij}(k); i, j \in \mathrm{E}, k \in \mathbb{N})$ defined by:

$$q_{ij}(k) = \mathbb{P}(J_{n+1} = j, L_{n+1} = k | J_n = i),\ k > 0, \text{ and } q_{ij}(0) = 0.$$

If (J,S) is a (homogeneous) Markov renewal chain, we can easily see that $(J_n)_{n \in \mathbb{N}}$ is a (homogeneous) Markov chain, called the embedded Markov chain (EMC) associated to the MRC (J,S). we denote by $\mathbf{P} = (p_{ij})_{i,j \in \mathrm{E}} \in \mathcal{M}_E$ the transition matrix of (J_n). Note also that, for any $i, j \in \mathrm{E}$, p_{ij} can be expressed in terms of the semi Markov kernel:

$$p_{ij} = \sum_{k=0}^{\infty} q_{ij}(k).$$

Let us introduce the cumulated semi-Markov kernel $\mathbf{Q} = (\mathbf{Q}(k), k \in \mathbb{N}) \in \mathcal{M}_{\mathrm{E}}(\mathbb{N})$ defined, for all $i, j \in \mathrm{E}$ and for all $k \in \mathbb{N}$, by

$$Q_{ij}(k) = \mathbb{P}(J_{n+1} = j, L_{n+1} \leq k | J_n = i) = \sum_{l=0}^{k} q_{ij}(l).$$

1.3.2 Definition of semi-Markov chains and sojourn time distributions

Definition 1.3.4. ***(Discrete-time semi-Markov chain)***

Let (J, S) be a Markov renewal chain. The chain $Z = (Z_k)_{k\in\mathbb{N}}$ is said to be a semi-Markov chain associated to the MRC (J, S) if

$$Z_k := J_{N(k)}, \quad k \in \mathbb{N},$$

where

$$N(k) := max\{n \geq 0, \;\; S_n \leq k\},$$

is the discrete-time counting process of the number of jumps in $[1, k] \subset \mathbb{N}$.

Thus Z_k gives the system state at time k. We have also $J_n = Z_{S_n}$ and $S_n = min\{k > S_{n-1} | Z_k \neq Z_{k-1}\}$, $n \in \mathbb{N}$.

Let the row vector $\alpha = (\alpha_1, \ldots, \alpha_s)$ denote the initial distribution of the semi Markov chain $Z = (Z_k)_{k\in\mathbb{N}}$ i.e $\alpha_i := \mathbb{P}(Z_0 = i) = \mathbb{P}(J_0 = i), \;\; i \in \mathrm{E}$.

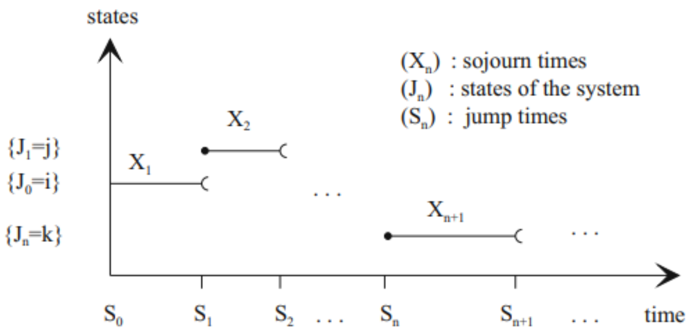


Figure 1.1: Sample path of a semi-Markov chain.

When investigating the evolution of a Markov renewal chain we are interested in two types of holding time distributions: the sojourn time distributions in a given state and the conditional distributions depending on the next state to be visited.

Definition 1.3.5. ***(Conditional distributions of sojourn times)***
For all $i, j \in \mathrm{E}$, let us define:

- *$f_{ij}(.)$, the conditional distribution of sojourn time in state i before going to state j:*

$$f_{ij}(k) := \mathbb{P}(L_{n+1} = k | J_n = i, J_{n+1} = j), \;\; \forall k \in \mathbb{N}.$$

- *$F_{ij}(.)$, the conditional cumulative distribution of L_{n+1}, $n \in \mathbb{N}$:*

$$F_{ij}(k) := \mathbb{P}(L_{n+1} \leq k | J_n = i, J_{n+1} = j) = \sum_{l=0}^{k} f_{ij}(l), \ \ \forall k \in \mathbb{N}.$$

Obviously, for all $i, j \in \mathrm{E}$ and for all $k \in \mathbb{N}$, we have:

$$f_{ij}(k) = q_{ij}(k)/p_{ij} \quad if \, p_{ij} \neq 0. \tag{1.5}$$

and, by convention, we put $f_{ij}(k) = \mathbb{1}_{\{k=\infty\}} \quad if \, p_{ij} = 0$.

Definition 1.3.6. ***(Sojourn time distributions in a given state)***
For all $i \in \mathrm{E}$, let us define:

- *$h_i(.)$, the sojourn time distribution in state i:*

$$h_i(k) := \mathbb{P}(L_{n+1} = k | J_n = i) = \sum_{j \in \mathrm{E}} q_{ij}(k), \forall k \in \mathbb{N}.$$

- *$H_i(.)$, the sojourn time cumulative distribution function in state i:*

$$H_i(k) := \mathbb{P}(L_{n+1} \leq k | J_n = i) = \sum_{l=1}^{k} h_i(l), \forall k \in \mathbb{N}.$$

- *$\overline{H}_i(.)$, the survival function of sojourn time in state i:*

$$\overline{H}_i(k) := \mathbb{P}(L_{n+1} > k | J_n = i) = 1 - \sum_{j \in \mathrm{E}} \sum_{n=1}^{k} q_{ij}(n), \forall k \in \mathbb{N}.$$

As we saw in equation 1.5 *the semi-Markov kernel introduced in Definition* 1.3.3 *verifies the relation:*
for all $i, j \in \mathrm{E}$ and $k \in \mathbb{N}$, we have $q_{ij}(k) = p_{ij} f_{ij}(k)$, such that $p_{ij} \neq 0$.

The following assumptions concerning the semi Markov chain will be needed in the rest of this work.

A1 The SMC is irreducible.

A2 The mean sojourn times are finite, i.e. $\sum\limits_{k=0} k h_i(k) < \infty$ for any state $i \in \mathrm{E}$.

A3 The Markov renewal process $(J_n, S_n)_{n \in \mathbb{N}}$ is aperiodic.

1.4 Elements of statistical estimation

Let us consider a sample path of the DTMRP $(J_n, S_n)_{n\in\mathbb{N}}$, censored at time $M \in \mathbb{N}$ ($L_{N(M)+1}$ is above u_M but it is unknown by how much), that is, a sequence of successively visited states and sojourn times:

$$\mathcal{H}(M) := (J_0, L_1, \ldots, J_{N(M)-1}, L_{N(M)}, J_{N(M)}, u_M),$$

where $N(M)$ is the number of jumps of the process in $[1, M] \subset \mathbb{N}$ and $u_M := M - S_{N(M)}$ is the censored sojourn time in the last visited state $J_{N(M)}$.

1.4.1 Empirical estimators

Taking a sample path $\mathcal{H}(M)$ of a DTMRP, for all $i, j \in E$ and $k \in \mathbb{N}$, $k \leq M$, we define the empirical estimators of the transition matrix of the embedded Markov chain p_{ij}, of the conditional sojourn time $f_{ij}(k)$ and of the discrete semi-Markov kernel $q_{ij}(k)$ by

$$\widehat{p}_{ij}(M) := \frac{N_{ij}(M)}{N_i(M)},\ \widehat{f}_{ij}(k, M) := \frac{N_{ij}(k, M)}{N_{ij}(M)},\ \widehat{q}_{ij}(k, M) := \frac{N_{ij}(k, M)}{N_i(M)}. \tag{1.6}$$

where $N_{ij}(k, M)$, $N_i(M)$ and $N_{ij}(M)$ are given by

- $N_i(M) := \sum_{n=1}^{N(M)} \mathbb{1}_{\{J_n=i\}}$: the number of visits to state i, up to time M;
- $N_{ij}(M) := \sum_{n=1}^{N(M)} \mathbb{1}_{\{J_{n-1}=i, J_n=j\}}$: the number of transitions from i to j, up to time M;
- $N_{ij}(k, M) := \sum_{n=1}^{N(M)} \mathbb{1}_{\{J_{n-1}=i, J_n=j, L_n=k\}}$: the number of transitions from i to j, up to time M, with sojourn time in state i equal to k, $1 \leq k \leq M$.

If $N_i(M) = 0$ we set $\hat{p}_{ij}(M) = 0$ and $\hat{q}_{ij}(k, M) = 0$ for any $k \in \mathbb{N}$, and if $N_{ij}(M) = 0$ we set $\hat{f}_{ij}(k, M) = 0$ for any $k \in \mathbb{N}$.

The likelihood function corresponding to the history $\mathcal{H}(M)$ is

$$\mathrm{L}(M) = \alpha_{J_0} \prod_{k=1}^{N(M)} p_{J_{k-1}J_k} f_{J_{k-1}J_k}(L_k) \overline{H}_{J_{N(M)}}(u_M),$$

where $\overline{H}_{J_{N(M)}}$ is the survival function in state i and α_i is the initial distribution of state i.

Lemma 1.4.1.1. *[4]*

For a semi-Markov chain $Z = (Z_n)_{n\in\mathbb{N}}$ we have

$$u_M/M \xrightarrow[M\to\infty]{a.s} 0.$$

The previous lemma tells us that, for large M, u_M does not add significant information to the likelihood function. For these reason, we will neglect the term $\overline{H}_{J_{N(M)}}(u_M)$ in the expression of the likelihood function $\mathrm{L}(M)$. On the other side, the sample path $\mathcal{H}(M)$ of the MRC $(J_n, S_n)_{n\in\mathbb{N}}$ contains only one observation of the initial distribution α of $(J_n)_{n\in\mathbb{N}}$, so the information on α_{J_0} does not increase with M. As we are interested in large-sample estimation of semi-Markov chains, the term α_{J_0} will be equally neglected in the expression of the likelihood function (see Billingsley, 1961a, page 4, for a similar discussion about Markov chain estimation).

For this reasons, instead of maximizing L(M) we will maximize the approached likelihood function defined by

$$\mathrm{L}_1(M) = \prod_{k=1}^{N(M)} p_{J_{k-1}J_k} f_{J_{k-1}J_k}(L_k). \tag{1.7}$$

And we will call the obtained estimators "approached maximum-likelihood estimators."

Proposition 1.1. *[4]*

For a sample path of a DTMRP $(J_n, S_n)_{n\in\mathbb{N}}$, of arbitrarily fixed length $M \in \mathbb{N}$, the empirical estimators $\widehat{p}_{ij}(M)$, $\widehat{f}_{ij}(k, M)$ and $\widehat{q}_{ij}(k, M)$, proposed in equation (1.6)*, are approached non-parametric maximum likelihood estimators i.e. they maximize the approached likelihood function* L_1*, given in equation* (1.7)*.*

Proof. We consider the approached likelihood function $\mathrm{L}_1(M)$ given by equation (1.7). Using the equality

$$\sum_{j=1}^{s} p_{ij} = 1, i \in \mathrm{E}. \tag{1.8}$$

the approached log-likelihood function can be written in the form:

$$\log(\mathrm{L}_1(M)) = \sum_{k=1}^{M} \sum_{i,j=1}^{s} [N_{ij}(M)\log(p_{ij}) + N_{ij}(k, M)\log(f_{ij}(k)) + \lambda_i(1 - \sum_{j=1}^{s} p_{ij})], \tag{1.9}$$

where the Lagrange multipliers λ_i are arbitrarily chosen constants.
In order to obtain the approached MLE of p_{ij}, we maximize equation (1.9) with respect to p_{ij}, and get $p_{ij} = N_{ij}(M)/\lambda_i$. Equation (1.8) becomes

$$1 = \sum_{j=1}^{s} p_{ij} = \sum_{j=1}^{s} \frac{N_{ij}(M)}{\lambda_i} = \frac{N_i(M)}{\lambda_i}.$$

Finally, we infer that the values λ_i which maximize equation (1.9) with respect to p_{ij} are given by $\lambda_i = N_i(M)$ and we obtain

$$\widehat{p}_{ij}(M) := \frac{N_{ij}(M)}{N_i(M)}.$$

The expression of $\widehat{f}_{ij}(k, M)$ can be obtained by the same method. Indeed, using the equality

$$\sum_{k=1}^{\infty} f_{ij}(k) = 1 \tag{1.10}$$

we write the approached log-likelihood function in the form:

$$\log(\mathrm{L}_1(M)) = \sum_{k=1}^{M} \sum_{i,j=1}^{s} [N_{ij}(M)\log(p_{ij}) + N_{ij}(k, M)\log(f_{ij}(k)) + \lambda_{ij}(1 - \sum_{k=1}^{\infty} f_{ij}(k))], \tag{1.11}$$

where λ_{ij} are arbitrarily chosen constants. Maximizing (1.11) with respect to $f_{ij}(k)$ we obtain $\widehat{f}_{ij}(k, M) := N_{ij}(k, M)/\lambda_{ij}$.

From Equation (1.10) we obtain $\lambda_{ij}(M) = N_{ij}(M)$. Thus
$\widehat{f}_{ij}(k, M) := N_{ij}(k, M)/N_{ij}(M)$.

In an analogous way we can prove that the expression of the approached MLE of the kernel $q_{ij}(k)$ is given by equation (1.6). □

Lemma 1.4.1. *[4] For a MRC that satisfies Assumptions A1 and A2, we have:*

1. $\lim_{M\to\infty} S_M = \infty$ *a.s;*

2. $\lim_{M\to\infty} N(M) = \infty$ *a.s.*

Lemma 1.4.2. *[4] For the DTMRP $(J_n, S_n)_{n\in\mathbb{N}}$. We have*

$$\frac{N_i(M)}{M} \xrightarrow[M\to\infty]{a.s} \frac{1}{\mu_{ii}}, \quad \frac{N_{ij}(M)}{M} \xrightarrow[M\to\infty]{a.s} \frac{p_{ij}}{\mu_{ii}}, \quad \frac{N(M)}{M} \xrightarrow[M\to\infty]{a.s} \frac{1}{\nu(l)\mu_{ll}}.$$

where μ_{ii} is the mean recurrence time of state i for the semi-Markov process $(Z_n)_{n\in\mathbb{N}}$, $(\nu(l); l \in \mathrm{E})$ the stationary distribution and l is an arbitrary fixed state.

1.5 Asymptotic properties of the estimators

In this section, we study the asymptotic properties (consistency and asymptotic normality) of the proposed estimators $\widehat{p}_{ij}(M)$, $\widehat{f}_{ij}(k,M)$ and $\widehat{q}_{ij}(k,M)$.

1.5.1 Strong consistency

Corollary 1.5.1. *[4] For any $i,j \in \mathrm{E}$, under A1, we have*

$$\widehat{p}_{ij}(M) = \frac{N_{ij}(M)}{N_i(M)} \xrightarrow[M\to\infty]{a.s} p_{ij}.$$

For $i,j \in \mathrm{E}$ two fixed states, let us also define the empirical estimator of the conditional cumulative distribution of $(L_n)_{n\in\mathbb{N}^*}$

$$\widehat{F}_{ij}(k,M) := \sum_{l=0}^{k} \widehat{f}_{ij}(l,M) = \sum_{l=0}^{k} \frac{N_{ij}(l,M)}{N_{ij}(M)}. \tag{1.12}$$

The following result concerns the convergence of $\widehat{f}_{ij}(k,M)$ and $\widehat{F}_{ij}(k,M)$.

Proposition 1.2. *[4] For any fixed arbitrary states $i,j \in \mathrm{E}$, the empirical estimators $\widehat{f}_{ij}(k,M)$ and $\widehat{F}_{ij}(k,M)$ proposed in equations (1.6) and (1.12), are uniformly strongly consistent, i.e.*

1. $\max\limits_{i,j\in\mathrm{E}} \max\limits_{0\le k\le M} |\widehat{F}_{ij}(k,M) \;-\; F_{ij}(k)| \xrightarrow[M\longrightarrow\infty]{a.s.} 0.$

2. $\max\limits_{i,j\in\mathrm{E}} \max\limits_{0\le k\le M} |\widehat{f}_{ij}(k,M) \;-\; f_{ij}(k)| \xrightarrow[M\longrightarrow\infty]{a.s.} 0.$

Proof. We first prove the strong consistency of the estimators using the SLLN theorem 1.1.1 . Second, we show the uniform consistency, i.e., that the convergence does not depend on the chosen k, $0 \le k \le M$. This second part is done by means of the Glivenko-Cantelli theorem 1.1.2.

Obviously, the strong consistency can be directly obtained using Glivenko-Cantelli theorem 1.1.2. Anyway, we prefer to derive separately the consistency result because it is easy and constructive.

Let us denote by $\{n_1, n_2, \ldots, n_{N_{ij}(M)}\}$ the transition times from state i to state j, up to time M. Note that we have

$$\widehat{F}_{ij}(k,M) = \frac{1}{N_{ij}(M)} \sum_{l=1}^{N_{ij}(M)} \mathbb{1}_{\{L_{n_l}\le k\}},$$

and

$$\widehat{f}_{ij}(k,M) = \frac{1}{N_{ij}(M)} \sum_{l=1}^{N_{ij}(M)} \mathbb{1}_{\{L_{n_l}=k\}}.$$

For any $l \in \{1, 2, \ldots, N_{ij}(M)\}$ we have

$$\mathbb{E}[\mathbb{1}_{\{L_{n_l}\leq k\}}] = \mathbb{P}(L_{n_l} \leq k) = F_{ij}(k),$$

and

$$\mathbb{E}[\mathbb{1}_{\{L_{n_l}=k\}}] = \mathbb{P}(L_{n_l} = k) = f_{ij}(k).$$

Since $N_{ij}(M) \xrightarrow[M\to\infty]{a.s} \infty$, applying the SLLN theorem 1.1.1 to the sequences of i.i.d. random variables $\{\mathbb{1}_{\{L_{n_l}\leq k\}}\}_{l\in\{1,2,\ldots,N_{ij}(M)\}}$ and $\{\mathbb{1}_{\{L_{n_l}=k\}}\}_{l\in\{1,2,\ldots,N_{ij}(M)\}}$, and using Theorem 1.1.3, we get

$$\widehat{F}_{ij}(k,M) = \frac{1}{N_{ij}(M)} \sum_{l=1}^{N_{ij}(M)} \mathbb{1}_{\{L_{n_l}\leq k\}} \xrightarrow[M\to\infty]{a.s} \mathbb{E}[\mathbb{1}_{\{L_{n_l}\leq k\}}] = F_{ij}(k),$$

and

$$\widehat{f}_{ij}(k,M) = \frac{1}{N_{ij}(M)} \sum_{l=1}^{N_{ij}(M)} \mathbb{1}_{\{L_{n_l}=k\}} \xrightarrow[M\to\infty]{a.s} \mathbb{E}[\mathbb{1}_{\{L_{n_l}=k\}}] = f_{ij}(k).$$

In order to obtain uniform consistency, from the Glivenko-Cantelli theorem 1.1.2, we have

$$\max_{0\leq k\leq m} |\frac{1}{m} \sum_{l=1}^{m} \mathbb{1}_{\{L_{n_l}\leq k\}} - F_{ij}(k)| \xrightarrow[M\to\infty]{a.s} 0.$$

Let us define $\xi_m := \max_{0\leq k\leq m} |\frac{1}{m} \sum_{l=1}^{m} \mathbb{1}_{\{L_{n_l}\leq k\}} - F_{ij}(k)|$. The previous convergence tells us that $\xi_m \xrightarrow[m\to\infty]{a.s} 0$. As $N(M) \xrightarrow[M\to\infty]{a.s} \infty$ (1.4.1) applying Theorem 1.1.3 we obtain $\xi_{N(M)} \xrightarrow[M\to\infty]{a.s} 0$ which reads

$$\max_{0\leq k\leq M} |\widehat{F}_{ij}(k,M) - F_{ij}(k)| \xrightarrow[M\to\infty]{a.s} 0.$$

As the state space E is finite, we take the maximum with respect to $i, j \in$ E and the desired result for $\widehat{F}_{ij}(k,M)$ follows.

Concerning the uniform consistency of $\widehat{f}_{ij}(k,M)$, note that we have

$$\max_{i,j\in \mathrm{E}} \max_{0\leq k\leq M} |\widehat{f}_{ij}(k,M) - f_{ij}(k)| = \max_{i,j\in \mathrm{E}} \max_{0\leq k\leq M} |\widehat{F}_{ij}(k,M) - \widehat{F}_{ij}(k-1,M) - F_{ij}(k) + F_{ij}(k-1)|$$

$$\leq \max_{i,j\in \mathrm{E}} \max_{0\leq k\leq M} |\widehat{F}_{ij}(k,M) - F_{ij}(k)| + \max_{i,j\in \mathrm{E}} \max_{0\leq k\leq M} |\widehat{F}_{ij}(k-1,M) - F_{ij}(k-1)|$$

and the result follows from the uniform strong consistency of $\widehat{F}_{ij}(k,M)$.□

Proposition 1.3. *[4] The empirical estimator of the semi-Markov kernel proposed in equation* (1.6) *is uniformly strongly consistent, i.e.*

$$\max_{i,j\in \mathrm{E}} \max_{0\le k\le M} |\widehat{q}_{ij}(k,M) - q_{ij}(k)| \xrightarrow[M\longrightarrow\infty]{a.s.} 0.$$

Proof. Firstly, from Corollary 1.5.1, we immediately obtain the almost sure convergence of $\widehat{p}_{ij}(M)$. The uniform strong consistency of $\widehat{q}_{ij}(k,M)$ follows from the consistency of the estimators $\widehat{p}_{ij}(M)$, $\widehat{f}_{ij}(k,M)$ (Proposition 1.1) and from the following inequality

$$\begin{aligned}
\max_{i,j\in \mathrm{E}} \max_{0\le k\le M} |\widehat{q}_{ij}(k,M) - q_{ij}(k)| &= \max_{i,j\in \mathrm{E}} \max_{0\le k\le M} |\widehat{p}_{ij}(M)\widehat{f}_{ij}(k,M) - \widehat{p}_{ij}(M) f_{ij}(k) \\
&\quad + \widehat{p}_{ij}(M) f_{ij}(k) - p_{ij} f_{ij}(k)| \\
&\le \max_{i,j\in \mathrm{E}} \widehat{p}_{ij}(M) \max_{i,j\in \mathrm{E}} \max_{0\le k\le M} |\widehat{f}_{ij}(k,M) - f_{ij}(k)| \\
&\quad + \max_{i,j\in \mathrm{E}} \max_{0\le k\le M} f_{ij}(k) \max_{i,j\in \mathrm{E}} |\widehat{p}_{ij}(M) - p_{ij}| \\
&\le \max_{i,j\in \mathrm{E}} |\widehat{p}_{ij}(M) - p_{ij}| + \max_{i,j\in \mathrm{E}} \max_{0\le k\le M} |\widehat{f}_{ij}(k,M) - f_{ij}(k)|.
\end{aligned}$$

The conclusion follows from the consistency of $\widehat{p}_{ij}(M)$ and $\widehat{f}_{ij}(k,M)$. □

1.5.2 Asymptotic normality

We present further theorem CLT for additive functionals of Markov renewal chains. Let f be a real function defined on $\mathrm{E}\times\mathrm{E}\times\mathbb{N}$. Define, for each $M\in\mathbb{N}$, the functional $W_f(M)$ as

$$W_f(M) := \sum_{n=1}^{N(M)} f(J_{n-1}, J_n, L_n),$$

or, equivalently,

$$W_f(M) := \sum_{i,j=1}^{s} \sum_{n=1}^{N_{ij}(M)} f(i,j,L_{ijn}),$$

where L_{ijn} is the n^{th} sojourn time of the chain in state i, before going to state j. Set

$$\begin{aligned}
A_{ij} &:= \sum_{x=1}^{\infty} f(i,j,x) q_{ij}(x), \qquad A_i := \sum_{j=1}^{s} A_{ij}, \\
B_{ij} &:= \sum_{x=1}^{\infty} f^2(i,j,x) q_{ij}(x), \qquad B_i := \sum_{j=1}^{s} B_{ij},
\end{aligned}$$

if the sums exist. Define

$$\begin{aligned}
r_i &:= \sum_{j=1}^{s} A_j \frac{\mu^*_{ii}}{\mu^*_{jj}}, \qquad m_f := \frac{r_i}{\mu_{ii}} \\
\sigma_i^2 &:= -r_i^2 + \sum_{j=1}^{s} B_j \frac{\mu^*_{ii}}{\mu^*_{jj}} + 2\sum_{r=1}^{s}\sum_{l\neq i}\sum_{k\neq i} A_{rl}A_k\mu^*_{ii}\frac{\mu^*_{li}+\mu^*_{ik}-\mu^*_{lk}}{\mu^*_{rr}\mu^*_{kk}},\ B_f := \frac{\sigma_i^2}{\mu_{ii}}
\end{aligned}$$

Where μ^*_{ii} is the mean recurrence time of state i for the Markov chain $(J_n)_{n\geq 0}$.

Theorem 1.5.1. *(Central Limit Theorem) [24]*

For an aperiodic Markov renewal chain that satisfies Assumptions A1 and A2 we have

$$\sqrt{M}\left[\frac{W_f(M)}{M} - m_f\right] \xrightarrow[M\to\infty]{\mathcal{D}} \mathcal{N}(0, B_f).$$

Theorem 1.5.2. *[4] For $i, j \in \mathrm{E}$, and $k \in \mathbb{N}$, $\sqrt{M}[\widehat{q}_{ij}(k, M) - q_{ij}(k)]$ converges in distribution, as $M \to \infty$, to a zero mean normal random variable with variance $\mu_{ii}q_{ij}(k)[1 - q_{ij}(k)]$.*

Proof. We present two different proofs of the theorem. The first one is based on the CLT for Markov renewal chains (Theorem 1.5.1). The second one relies on the Lindeberg-Lévy CLT for martingales (Theorem 1.1.4).

Method 1.

$$\begin{aligned}
\sqrt{M}[\widehat{q}_{ij}(k,M) - q_{ij}(k)] &= \frac{M}{N_i(M)}\frac{1}{\sqrt{M}}\sum_{n=1}^{N(M)}[\mathbb{1}_{\{J_n=j,L_n=k\}} - q_{ij}(k)]\mathbb{1}_{\{J_{n-1}=i\}} \\
&= \sum_{n=1}^{N(M)} f(J_{n-1}, J_n, L_n).
\end{aligned}$$

Let us consider the function

$$f(m, l, u) := \mathbb{1}_{\{m=i,l=j,u=k\}} - q_{ij}(k)\mathbb{1}_{\{m=i\}}.$$

Using the notation from the Pyke and Schaufele's CLT, we have

$$W_f(M) = \sum_{n=1}^{N(M)} f(J_{n-1}, J_n, L_n) = \sum_{n=1}^{N(M)}[\mathbb{1}_{\{J_n=j,L_n=k\}} - q_{ij}(k)]\mathbb{1}_{\{J_{n-1}=i\}}.$$

In order to apply Pyke and Schaufeles'central limit theorem for Markov renewal processes (Theorem 1.5.1), we need to compute $A_{ml}, A_m, B_{ml}, B_m, m_f$ and B_f for $m, l \in \mathrm{E}$.

$$\begin{aligned}
A_{ml} &:= \sum_{u=1}^{\infty} f(m,l,u) q_{ml}(u), \\
&:= \sum_{u=1}^{\infty} \mathbb{1}_{\{m=i, l=j, u=k\}} q_{ml}(u) - \sum_{u=1}^{\infty} \mathbb{1}_{\{m=i\}} q_{ij}(k) q_{ml}(u) \\
&:= \delta_{mi}\delta_{lj} \sum_{u=1}^{\infty} \mathbb{1}_{\{u=k\}} q_{ij}(u) - \delta_{mi} q_{ij}(k) \sum_{u=1}^{\infty} q_{il}(u) = q_{ij}(k)\delta_{mi}(\delta_{lj} - p_{il}) \\
A_m &:= \sum_{l=1}^{s} A_{ml} = q_{ij}(k)\delta_{mi}[\sum_{l=1}^{s} \delta_{lj} - \sum_{l=1}^{s} p_{il}] = 0. \\
B_{ml} &:= \sum_{u=1}^{\infty} f^2(m,l,u) q_{ml}(u) \\
&:= \sum_{u=1}^{\infty} \mathbb{1}_{\{m=i, l=j, u=k\}} q_{ml}(u) + \sum_{u=1}^{\infty} \mathbb{1}_{\{m=i\}} q_{ij}^2(k) q_{ml}(u) \\
&\quad -2 \sum_{u=1}^{\infty} \mathbb{1}_{\{m=i, l=j, u=k\}} q_{ij}(k) q_{ml}(u) \\
&:= q_{ij}(k)\delta_{mi}\delta_{lj} + q_{ij}^2(k)\delta_{mi}p_{il} - 2q_{ij}^2(k)\delta_{mi}\delta_{lj} \\
B_m &:= \sum_{l=1}^{s} B_{ml} = \delta_{mi} q_{ij}(k)[1 - q_{ij}(k)].
\end{aligned}$$

Finally, we obtain

$$\begin{aligned}
r_i &:= \sum_{m=1}^{s} A_m \frac{\mu_{ii}^*}{\mu_{mm}^*} = 0, & m_f &:= \frac{r_i}{\mu_{ii}} = 0, \\
\sigma_i^2 &:= \sum_{m=1}^{s} B_m \frac{\mu_{ii}^*}{\mu_{mm}^*} = q_{ij}(k)[1 - q_{ij}(k)], & B_f &:= \frac{\sigma_i^2}{\mu_{ii}} = \frac{q_{ij}(k)[1 - q_{ij}(k)]}{\mu_{ii}}.
\end{aligned}$$

Since $N_i(M)/M \underset{M\to\infty}{\xrightarrow{a.s}} 1/\mu_{ii}$ (see Lemma 1.4.2), we conclude as follows:

$$\sqrt{M}[\widehat{q}_{ij}(k,M) - q_{ij}(k)] \underset{M\to\infty}{\xrightarrow{\mathcal{D}}} \mathcal{N}(0, \mu_{ii} q_{ij}(k)[1 - q_{ij}(k)]).$$

Method 2.

For $i, j \in \mathrm{E}$ arbitrarily fixed states and $k \in \mathbb{N}$ arbitrarily fixed positive integer, we write the random variable $\sqrt{M}[\widehat{q}_{ij}(k,M) - q_{ij}(k)]$ as

$$\sqrt{M}[\widehat{q}_{ij}(k,M) - q_{ij}(k)] = \frac{M}{N_i(M)} \frac{1}{\sqrt{M}} \sum_{n=1}^{N(M)} \left[\mathbb{1}_{\{J_n=j, L_n=k\}} - q_{ij}(k) \right] \mathbb{1}_{\{J_{n-1}=i\}}.$$

Let $\mathcal{F}_n$ be the σ-algebra defined by $\mathcal{F}_n := \sigma(J_l, L_l; l \leq n), n \geq 0$, and let Y_n be the random variable

$$Y_n = 1\!\!1_{\{J_{n-1}=i, J_n=j, L_n=k\}} - q_{ij}(k) 1\!\!1_{\{J_{n-1}=i\}}.$$

Obviously, Y_n is $\mathcal{F}_n$-measurable and $\mathcal{F}_n \subseteq \mathcal{F}_{n+1}$, for all $n \in \mathbb{N}$. Moreover, we have

$$\begin{aligned}
\mathbb{E}(Y_n \mid \mathcal{F}_{n-1}) &= \mathbb{P}(J_{n-1}=i, J_n=j, L_n=k \mid \mathcal{F}_{n-1}) - q_{ij}(k)\mathbb{P}(J_{n-1}=i \mid \mathcal{F}_{n-1}) \\
&= 1\!\!1_{\{J_{n-1}=i\}}\mathbb{P}(J_n=j, L_n=k \mid J_{n-1}=i) - q_{ij}(k)1\!\!1_{\{J_{n-1}=i\}} \\
&= 0.
\end{aligned}$$

Therefore, $(Y_n)_{n\in\mathbb{N}}$ is an $\mathcal{F}_n$-martingale difference and $(\sum_{l=1}^{n} Y_l)_{l\in\mathbb{N}}$ is an $\mathcal{F}_n$-martingale. Note also that, as Y_l is bounded for all $l \in \mathbb{N}$, we have

$$\frac{1}{\sqrt{n}} \sum_{l=1}^{n} \mathbb{E}(Y_l^2 1\!\!1_{\{|Y_l|>\epsilon\sqrt{n}\}}) \underset{n\to\infty}{\longrightarrow} 0.$$

For any $\epsilon > 0$. Using the CLT for martingales (Theorem 1.1.4) we obtain

$$\frac{1}{\sqrt{n}} \sum_{l=1}^{n} Y_l \xrightarrow[n\to\infty]{\mathcal{D}} \mathcal{N}(0, \sigma^2), \tag{1.13}$$

where $\sigma^2 > 0$ is given by

$$\sigma^2 = \lim_{n\to\infty} \frac{1}{\sqrt{n}} \sum_{l=1}^{n} \mathbb{E}(Y_l^2 \mid \mathcal{F}_{l-1}) > 0.$$

As $N(M)/M \xrightarrow[M\to\infty]{a.s} 1/\nu(l)\mu_{ll}$ applying Anscombe's theorem (Theorem 1.1.5) we obtain

$$\frac{1}{\sqrt{N(M)}} \sum_{l=1}^{N(M)} Y_l \xrightarrow[M\to\infty]{\mathcal{D}} \mathcal{N}(0, \sigma^2). \tag{1.14}$$

To obtain σ^2, we need to compute Y_l^2 and $\mathbb{E}(Y_l^2 \mid \mathcal{F}_{l-1})$. First,

$$Y_l^2 = 1\!\!1_{\{J_{l-1}=i, J_l=j, L_l=k\}} + (q_{ij}(k))^2 1\!\!1_{\{J_{l-1}=i\}} - 2q_{ij}(k) 1\!\!1_{\{J_{l-1}=i, J_l=j, L_l=k\}}.$$

Second,

$$\begin{aligned}
\mathbb{E}(Y_l^2 \mid \mathcal{F}_{l-1}) &= 1\!\!1_{\{J_{l-1}=i\}}\mathbb{P}(J_l=j, L_l=k \mid J_{l-1}=i) \\
&\quad + (q_{ij}(k))^2 1\!\!1_{\{J_{l-1}=i\}} - 2 1\!\!1_{\{J_{l-1}=i\}} q_{ij}(k)\mathbb{P}(J_l=j, L_l=k \mid J_{l-1}=i) \\
&= 1\!\!1_{\{J_{l-1}=i\}} q_{ij}(k) + (q_{ij}(k))^2 1\!\!1_{\{J_{l-1}=i\}} - 2(q_{ij}(k))^2 1\!\!1_{\{J_{l-1}=i\}} \\
&= 1\!\!1_{\{J_{l-1}=i\}} q_{ij}(k)[1 - q_{ij}(k)].
\end{aligned}$$

Thus, σ^2 given by

$$\sigma^2 = \lim_{n\to\infty} (\frac{1}{\sqrt{n}} \sum_{l=1}^{n} \mathbb{1}_{\{J_{l-1}=i\}}) q_{ij}(k)[1 - q_{ij}(k)] = \nu(i) q_{ij}(k)[1 - q_{ij}(k)],$$

where ν is the stationary distribution of the embedded Markov chain $(J_n)_{n\in\mathbb{N}}$. The random variable of interest $\sqrt{M}[\widehat{q}_{ij}(k,M) - q_{ij}(k)]$ can be written as

$$\begin{aligned}\sqrt{M}[\widehat{q}_{ij}(k,M) - q_{ij}(k)] &= \frac{M}{N_i(M)} \frac{1}{\sqrt{M}} \sqrt{N(M)} \frac{1}{\sqrt{N(M)}} \sum_{l=1}^{N(M)} Y_l \\ &= \frac{M}{N_i(M)} \sqrt{\frac{N(M)}{M}} \frac{1}{\sqrt{N(M)}} \sum_{l=1}^{N(M)} Y_l.\end{aligned}$$

Note that we have

$$\begin{aligned}\frac{N_i(M)}{M} &\xrightarrow[M\to\infty]{a.s} \frac{1}{\mu_{ii}}, \\ \frac{N(M)}{M} &\xrightarrow[M\to\infty]{a.s} \frac{1}{\nu(i)\mu_{ii}}.\end{aligned}$$

Using these results and convergence (1.14), we obtain that $\sqrt{M}[\widehat{q}_{ij}(k,M) - q_{ij}(k)]$ converges in distribution, as M tends to infinity, to a zero-mean normal random variable, of variance

$$\begin{aligned}\sigma_q^2(i,j,k) &= (\mu_{ii}\sqrt{1/\mu_{ii}\nu(i)})^2 \nu(i) q_{ij}(k)[1 - q_{ij}(k)] \\ &= \mu_{ii} q_{ij}(k)[1 - q_{ij}(k)],\end{aligned}$$

which is the desired result. □

Chapter 2

Hidden Markov and semi Markov models

2.1 Introduction

Many of the most powerful sequence analysis methods are now based on principles of probabilistic modeling, such as Hidden Markov Models (HMMs) and Hidden Semi Markov Models (HSMMs).

The basic idea of a hidden model is the following: we observe the evolution in time of a certain phenomenon (observed process), but we are interested in the evolution of another phenomenon, which we are not able to observe (hidden process). The two processes are related in the sense that the state occupied by the observed process depends on the state that the hidden process is in. To get one of the most intuitive and general examples of a hidden model, one can think the observed process as a received signal and the hidden process as the emitted signal.
Since being introduced by Baum and Petrie (1966) [5], the HMMs have become very popular in a wide range of applications like biology [12][14], speech recognition [26], image processing, and text recognition.

The main drawback of hidden Markov models comes from the Markov property, which requires that the sojourn time in a state be geometrically distributed. This makes the hidden Markov models too restrictive from a practical point of view. In order to solve this kind of problem in the field of speech recognition, Ferguson (1980) [15] proposed a model that allows arbitrary sojourn time distributions for the hidden process. This is called a hidden semi-Markov model, which is extended from hidden Markov models.

Combining the flexibility of the semi-Markov processes with the proved advantages of HMMs, we obtain HSMMs, which are a powerful tool for applications and offer a rich statistical framework.

2.2 Hidden Markov model

Example 1. *Let us consider the scenario below where the weather (the hidden variable), can be hot, mild or cold and the observed variables are the type of clothing worn. The arrows represent transitions from a hidden state to another hidden state or from a hidden state to an observed variable. Notice that, true to the Markov assumption, each state only depends on the previous state and not on any other prior states.*

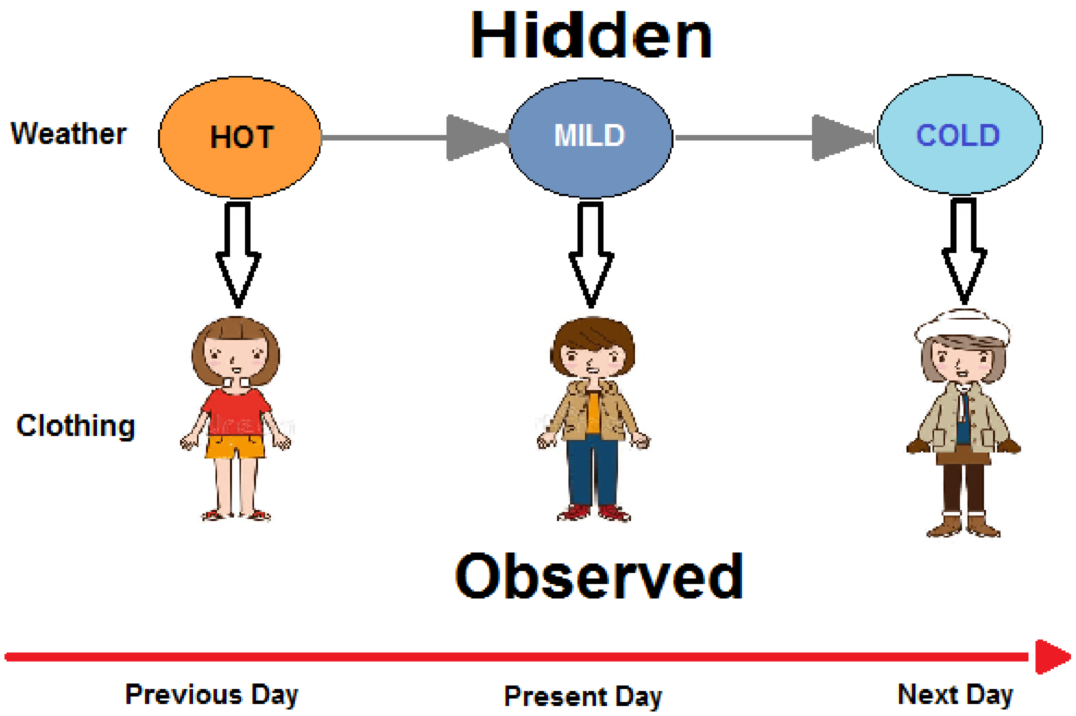
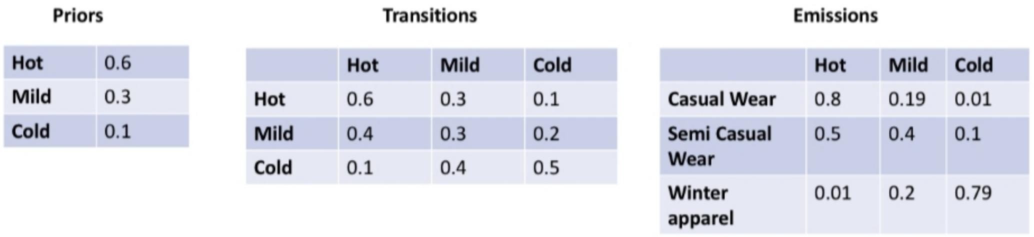


Figure 2.1: Example of Hidden Markov model.

The following tables represent the initial, transition, emission probabilities:



Priors

Hot	0.6
Mild	0.3
Cold	0.1

Transitions

	Hot	Mild	Cold
Hot	0.6	0.3	0.1
Mild	0.4	0.3	0.2
Cold	0.1	0.4	0.5

Emissions

	Hot	Mild	Cold
Casual Wear	0.8	0.19	0.01
Semi Casual Wear	0.5	0.4	0.1
Winter apparel	0.01	0.2	0.79

Generally, the term "states" are used to refer to the hidden states and "observations" are used to refer to the observed states.

Once we know the joint probability of a sequence of hidden states, we determine the best possible sequence i.e. the sequence with the highest probability and choose that sequence as the best sequence of hidden states.

In order to compute the joint probability of a sequence of hidden states, we need to assemble three types of information:

1. ***Transition probability*** *-the probability of transitioning to a new state conditioned on a present state.*

2. ***Emission probability*** *-the probability of transitioning to an observed conditioned on a hidden state.*

3. ***Initial state probability*** *-the initial probability of transitioning to a hidden state. This can also be looked at as the prior probability.*

The above information can be computed directly from our training data. In the case of our weather example in ***Figure*** *2.1, our training data would consist of the hidden state and observations for a number of days. We could build our matrices of: transitions, emission and initial state probabilities directly from this training data.*

The example tables show a set of possible values that could be derived for the weather/clothing scenario.

Remark 2.2.1. *In our work, we only consider the case when the observations were represented as discrete symbols chosen from a finite set* E*, and therefore we could use a discrete probability density within each state as this model.*

To define an HMM, we need some elements:

- The number of the hidden states is N. We denote these states by N : $\{s_1, s_2, \ldots, s_N\}$.
- The number of the observable states is M. We denote them by M : $\{r_1, r_2, \ldots, r_M\}$.
- $(X_n : n = 0, 1, \ldots)$ is an unobserved Markov chain with transition probability matrix $\mathbf{A} := (p_{ij})_{1\leq i,j\leq N}$, and an initial state distribution $\alpha = (\alpha_i)$.
- For a finite observation sequence $(Y_n : n = 0, 1, \ldots, T)$, where T is any fixed number, we have a fundamental assumption connecting the hidden state sequence $(X_n : n = 0, 1, \ldots, T)$ and the observation sequence, that is statistical independence of observations $(Y_n : n = 0, 1, \ldots, T)$. If we formulate this assumption mathematically, we have

$$\begin{aligned}\mathbb{P}(Y_0 = r_{j_0}, Y_1 = r_{j_1}, \ldots, Y_T = r_{j_T} | X_0 = s_{i_0}, X_1 = s_{i_1}, \ldots, X_T = s_{i_T})\\ = \prod_{n=0}^{T} \mathbb{P}(Y_n = r_{j_n} | X_n = s_{i_n}). \quad (2.1)\end{aligned}$$

where $1 \leq i_0, i_1, \ldots, i_T \leq N$, $1 \leq j_0, j_1, \ldots, j_T \leq M$. To simplify the notation, we denote the event sequence $(s_{i_0}, s_{i_1}, \ldots, s_{i_T})$ by $\mathbf{s_0^T}$, $(r_{i_0}, r_{i_1}, \ldots, r_{i_T})$ by $\mathbf{r_0^T}$, and denote $(X_0, X_1, \ldots, X_T)$ by $\mathbf{X_0^T}$, $(Y_0, Y_1, \ldots, Y_T)$ by $\mathbf{Y_0^T}$. Then, we put

$$b_j(k) = \mathbb{P}(Y_n = r_k | X_n = s_j), 1 \leq j \leq N, 1 \leq k \leq M, t = 0, 1, 2, \ldots$$

We may rewrite the formula 2.1 by

$$\mathbb{P}(\mathbf{Y_0^T} = \mathbf{r_0^T} | \mathbf{X_0^T} = \mathbf{s_0^T}) = \prod_{n=0}^{T} b_{i_n}(j_n) \quad (2.2)$$

By the knowledge of Markov chain, we know $\mathbb{P}(\mathbf{X_0^T} = \mathbf{s_0^T})$, the probability of the state sequence $\mathbf{s_0^T}$,

$$\begin{aligned}\mathbb{P}(\mathbf{X_0^T} = \mathbf{s_0^T}) &= \mathbb{P}(X_0 = s_{i_0}, X_1 = s_{i_1}, \ldots, X_T = s_{i_T})\\ &= \alpha_{i_0} p_{i_0 i_1} \ldots p_{i_{T-1} i_T}\\ &= \prod_{n=0}^{T} p_{i_{n-1} i_n},\end{aligned}$$

where $p_{i_{-1} i_0} = \alpha_{i_0}$. Hence, the joint probability of $\mathbf{s_0^T}$ and $\mathbf{r_0^T}$ is

$$\begin{aligned}\mathbb{P}(\mathbf{X_0^T} = \mathbf{s_0^T}, \mathbf{Y_0^T} = \mathbf{r_0^T}) &= \mathbb{P}(\mathbf{Y_0^T} = \mathbf{r_0^T} | \mathbf{X_0^T} = \mathbf{s_0^T}).\mathbb{P}(\mathbf{X_0^T} = \mathbf{s_0^T})\\ &= \prod_{n=0}^{T} [b_{i_n}(j_n) p_{i_{n-1} i_n}]. \quad (2.3)\end{aligned}$$

The transition probability matrix $\mathbf{A}$, the initial state distribution α and the matrix $\mathbf{B} = [b_j(k)], 1 \leq j \leq N, 1 \leq k \leq M$, define a hidden Markov model completely.

Therefore we can use a compact notation $\lambda = (\mathbf{A}, \mathbf{B}, \alpha)$ to denote a hidden Markov model with discrete probability distribution. We may think of λ as a parameter of the hidden Markov model. We denote the probability given a model λ by P_λ later.

So far, we know a hidden Markov model has several components. It has a set of states $s_1, s_2, \ldots, s_N$, a set of output symbols $r_1, r_2, \ldots, r_M$, a set of transitions which have associated with them a probability and an output symbol, and a starting state. When a transition is taken, it produces an output symbol. The complicating factor is that the output symbol given is not necessarily unique to that transition, and thus it is difficult to determine which transition was the one actually taken and this is why they are termed "hidden".

Lemma 2.2.1.

$$P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}) = \sum_{1 \leq i_0, i_1, \ldots, i_T \leq N} \prod_{n=0}^{T} p_{i_{n-1} i_n} b_{i_n}(j_n).$$

Proof. We want to compute $P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})$, the probability of the observation sequence given the model λ. From time 0 to time T, we consider every possible hidden state sequence $\mathbf{s_0^T}$. Then the probability of $\mathbf{r_0^T}$ is obtained by summing the joint probability over $\mathbf{r_0^T}$ and all possible $\mathbf{s_0^T}$, that is

$$P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})$$

$$= \sum_{\textit{all possible } \mathbf{s_0^T}} P_\lambda(\mathbf{X_0^T} = \mathbf{s_0^T}, \mathbf{Y_0^T} = \mathbf{r_0^T})$$

$$= \sum_{1 \leq i_0, i_1, \ldots, i_T \leq N} \prod_{n=0}^{T} [p_{i_{n-1} i_n} b_{i_n}(j_n)].$$

Lemma 2.2.1 gives us a method to compute the probability of a sequence of observations $\mathbf{r_0^T}$. But unfortunately, this calculation is computationally unfeasible. Because this formula involves on the order of $(T+1) \cdot N^{(T+1)}$ calculations. We are going to introduce some more efficient methods in the next section.

We can understand Lemma 2.2.1 from another point of view. A hidden Markov model consists of a set of hidden states $s_1, s_2, \ldots, s_N$ connected by directed edges. Each state assigns probabilities to the characters of the alphabet used in the observable sequence and to the edges leaving the state.

A path in an HMM, $s_{i_0}, s_{i_1}, \ldots, s_{i_T}$, is a sequence of states such that there is an edge from each state in the path to the next state in the path. And the probability of this path is the product of the probabilities of the edges traversed, that is $P(\mathbf{X_0^T} = \mathbf{s_0^T})$.

Each path through the HMM gives a probability distribution for each position in a string of the same length, based on the probabilities for the characters in the corresponding states. The probability of the observable sequence given a particular path is the product of the probabilities of the characters, that is

$$\mathbb{P}(\mathbf{Y_0^T} = \mathbf{r_0^T} | \mathbf{X_0^T} = \mathbf{s_0^T}) = \prod_{n=0}^{T} b_{i_n}(j_n).$$

The probability of any sequence of characters is the sum, over all paths whose length is the same as the sequence, of the probability of the path times the probability of the sequence given the path, that is the result of Lemma 2.2.1, as it shown in **Figure** 2.2 .

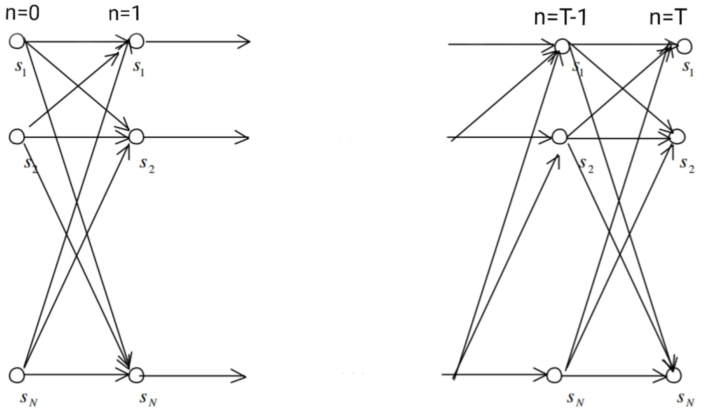


Figure 2.2: Paths of an HMM.

A Hidden Markov Model has a very similar property as a Markov process, that is given the values of X_n, the values of $Y_s, s \geq n$, do not depend on the values of $X_u, u < n$. The probability of any particular future observation of the model when its present hidden state is known exactly, is not altered by additional knowledge concerning its past hidden behavior. In formal terms, we have Lemma 2.2.2.

Lemma 2.2.2.

$$P_\lambda(Y_u = r_{j_u} | \mathbf{X_0^n} = \mathbf{s_0^n}) = P_\lambda(Y_u = r_{j_u} | X_n = s_{i_n}),\ u \geq n.$$

Proof. Firstly, we prove this result is true when $u = n$.

$$\begin{aligned}
P_\lambda(Y_n = r_{j_n}|\mathbf{X_0^n} = \mathbf{s_0^n}) &= \sum_{0\le j_0,\ldots,j_{n-1}\le M} P_\lambda(\mathbf{Y_0^n} = \mathbf{r_0^n}|\mathbf{X_0^n} = \mathbf{s_0^n}) \\
&= \sum_{0\le j_0,\ldots,j_{n-1}\le M} P_\lambda(\mathbf{Y_0^{n-1}} = \mathbf{r_0^{n-1}}|\mathbf{X_0^{n-1}} = \mathbf{s_0^{n-1}}) \cdot P_\lambda(Y_n = r_{j_n}|X_n = s_{i_n}) \\
&= 1 \cdot P_\lambda(Y_n = r_{j_n}|X_n = s_{i_n}) \\
&= P_\lambda(Y_n = r_{j_n}|X_n = s_{i_n})
\end{aligned}$$

Specially, if we have $q \le n$, we will have

$$\begin{aligned}
P_\lambda(Y_n = r_{j_n}|\mathbf{X_q^n} = \mathbf{s_q^n}) &= \sum_{0\le j_q,\ldots,j_{n-1}\le M} P_\lambda(\mathbf{Y_q^n} = \mathbf{r_q^n}|\mathbf{X_q^n} = \mathbf{s_q^n}) \\
&= \sum_{0\le j_q,\ldots,j_{n-1}\le M} P_\lambda(\mathbf{Y_q^{n-1}} = \mathbf{r_q^{n-1}}|\mathbf{X_q^{n-1}} = \mathbf{s_q^{n-1}}) \cdot P_\lambda(Y_n = r_{j_n}|X_n = s_{i_n}) \\
&= 1 \cdot P_\lambda(Y_n = r_{j_n}|X_n = s_{i_n}) \\
&= P_\lambda(Y_n = r_{j_n}|X_n = s_{i_n})
\end{aligned}$$

Then, we prove it is also true when $u > n$.

$$P_\lambda(Y_u = r_{j_u}|\mathbf{X_0^n} = \mathbf{s_0^n})$$

$$\begin{aligned}
&= \sum_{0\le j_0,\ldots,j_{u-1}\le M} P_\lambda(\mathbf{Y_0^u} = \mathbf{r_0^u}|\mathbf{X_0^n} = \mathbf{s_0^n}) \\
&= \sum_{0\le j_0,\ldots,j_{u-1}\le M} \sum_{0\le i_{n+1},\ldots,i_u\le N} P_\lambda(\mathbf{Y_0^u} = \mathbf{r_0^u}, \mathbf{X_{n+1}^u} = \mathbf{s_{n+1}^u})|\mathbf{X_0^n} = \mathbf{s_0^n}) \\
&= \sum_{0\le j_0,\ldots,j_{u-1}\le M} \sum_{0\le i_{n+1},\ldots,i_u\le N} P_\lambda(\mathbf{Y_0^u} = \mathbf{r_0^u}|\mathbf{X_0^u} = \mathbf{s_0^u}) \cdot P_\lambda(\mathbf{X_{n+1}^u} = \mathbf{s_{n+1}^u}|X_n = s_{i_n}) \\
&= \sum_{0\le j_0,\ldots,j_{u-1}\le M} \sum_{0\le i_{n+1},\ldots,i_u\le N} P_\lambda(\mathbf{Y_0^{u-1}} = \mathbf{r_0^{u-1}}|\mathbf{X_0^{u-1}} = \mathbf{s_0^{u-1}}) \cdot P_\lambda(Y_u = r_{j_u}|X_u = s_{i_u}) \\
&\quad \cdot P_\lambda(\mathbf{X_{n+1}^u} = \mathbf{s_{n+1}^u}|X_n = s_{i_n})
\end{aligned}$$

$$\begin{aligned}
&= \sum_{0\le i_{n+1},\ldots,i_u\le N} P_\lambda(Y_u = r_{j_u}|X_u = s_{i_u}) \cdot P_\lambda(\mathbf{X_{n+1}^u} = \mathbf{s_{n+1}^u}|X_n = s_{i_n}) \\
&= P_\lambda(Y_u = r_{j_u}|X_n = s_{i_n})
\end{aligned}$$

Lemma 2.2.2 will be widely used in next section to help solve the basic problems of HMM.

Remark 2.2.2. $(Y_n, n \geq 0)$ *are not independent.*

Proof.

We take $T = 1$. From Lemma 2.2.1, we have

$$P_\lambda(\mathbf{Y_0^1} = \mathbf{r_0^1}) = \sum_{i_0,i_1=1}^{N} b_{i_0}(j_0) b_{i_1}(j_1) \alpha_{i_0} p_{i_0 i_1}.$$

But $P_\lambda(Y_0 = r_{j_0}) = \sum_{i_0=1}^{N} b_{i_0}(j_0)\alpha_{i_0}$, and

$$P_\lambda(Y_1 = r_{j_1}) = \sum_{i_1}^{N} P_\lambda(Y_1 = r_{j_1} | X_1 = s_{i_1}) P_\lambda(X_1 = s_{i_1})$$

$$= \sum_{i_1}^{N} b_{i_1}(j_1)[\sum_{i_0}^{N} p_{i_0 i_1}\alpha_{i_0}] = \sum_{i_0,i_1=1}^{N} b_{i_1}(j_1) p_{i_0 i_1}\alpha_{i_0}.$$

Hence $P_\lambda(\mathbf{Y_0^1} = \mathbf{r_0^1}) \neq P_\lambda(Y_0 = r_{j_0}) \cdot P_\lambda(Y_1 = r_{j_1})$. Therefore, the sequence $(Y_n, n \geq 0)$ are not independent. Actually, it is very natural to be understood. Because for each Y_n, it is generated by the corresponding X_n, and the hidden sequence $(X_n, n = 0, 1, \ldots)$ are not independent.

There is a very easy example. We take $N = M$ and $b_i(j) = \delta_{ij}$. Then Y_n is a Markov chain.

2.2.1 Three basic problems of HMM and their solutions

Once we have an HMM, there are three problems of interest.

1.The Evaluation Problem

Given a hidden Markov model λ and a sequence of observations $\mathbf{Y_0^T} = \mathbf{r_0^T}$ what is the probability that the observations are generated by the model, i.e. $P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})$. We can also view the problem as how well a given model matches a given observation sequence. By the second viewpoint, if we have several competing models, the solution to the evaluation problem will give us a best model which best matches the observation sequence.

The most straightforward way of doing this is using Lemma 2.2.1. But it involves a lot of calculations. The more efficient methods are called the forward procedure and the backward procedure [26]. We will introduce these two procedures first, then reveal the mathematical idea inside them.

Definition 2.2.1. ***(The Forward Procedure)***

Fix $\mathbf{r_0^n}$ and consider the forward variable $\gamma_n(i_n)$ defined as

$$\gamma_n(i_n) = P_\lambda(\mathbf{Y_0^n} = \mathbf{r_0^n}, X_n = s_{i_n}), \quad n = 0, 1, \ldots, T, \quad 1 \leq i_n \leq N.$$

that is the probability of the partial observation sequence, $r_{j_0}, r_{j_1}, \ldots, r_{j_n}$(until time n) and at time n the hidden state is s_{i_n}. So for the forward variable $\gamma_n(i_n)$, we only consider those paths which end at the state s_{i_n} at the time n. We can solve for $\gamma_n(i_n)$ inductively, as follows:

1. *Initialization: For $n = 0$,*

$$\gamma_0(i_0) = \alpha_{i_0} b_{i_0}(j_0), \quad 1 \leq i_0 \leq N.$$

2. *Induction $(n = 0, 1, \ldots, T-1)$:*

$$\begin{aligned}
&\gamma_{n+1}(i_{n+1}) \\
&= P_\lambda(\mathbf{Y_0^{n+1}} = \mathbf{r_0^{n+1}}, X_{n+1} = s_{i_{n+1}}) \\
&= \sum_{0 \leq i_0, \ldots, i_n \leq N} P_\lambda(\mathbf{Y_0^{n+1}} = \mathbf{r_0^{n+1}}, \mathbf{X_0^{n+1}} = \mathbf{s_0^{n+1}}) \\
&= \sum_{0 \leq i_0, \ldots, i_n \leq N} P_\lambda(\mathbf{Y_0^{n+1}} = \mathbf{r_0^{n+1}} | \mathbf{X_0^{n+1}} = \mathbf{s_0^{n+1}}) \cdot P_\lambda(\mathbf{X_0^{n+1}} = \mathbf{s_0^{n+1}}) \\
&= \sum_{0 \leq i_0, \ldots, i_n \leq N} P_\lambda(\mathbf{Y_0^n} = \mathbf{r_0^n} | \mathbf{X_0^n} = \mathbf{s_0^n}) \cdot P_\lambda(Y_{n+1} = r_{j_{n+1}} | X_{n+1} = s_{i_{n+1}}) \\
&\quad \cdot P_\lambda(X_{n+1} = s_{i_{n+1}} | \mathbf{X_0^n} = \mathbf{r_0^n}) \cdot P_\lambda(\mathbf{X_0^n} = \mathbf{r_0^n}) \\
&= [\sum_{0 \leq i_0, \ldots, i_n \leq N} P_\lambda(\mathbf{Y_0^n} = \mathbf{r_0^n}, \mathbf{X_0^n} = \mathbf{s_0^n}) \cdot P_\lambda(X_{n+1} = s_{i_{n+1}} | X_n = s_{i_n})] \\
&\quad \cdot P_\lambda(Y_{n+1} = r_{j_{n+1}} | X_{n+1} = s_{i_{n+1}}) \\
&= [\sum_{0 \leq i_n \leq N} P_\lambda(\mathbf{Y_0^n} = \mathbf{r_0^n}, X_n = s_{i_n}) \cdot P_\lambda(X_{n+1} = s_{i_{n+1}} | X_n = s_{i_n})] \\
&\quad \cdot P_\lambda(Y_{n+1} = r_{j_{n+1}} | X_{n+1} = s_{i_{n+1}}) \\
&= [\sum_{i_n=1}^{N} \gamma_n(i_n) p_{i_n i_{n+1}}] b_{i_{n+1}}(j_{n+1})
\end{aligned}$$

3. *Termination:*

$$\begin{aligned}
P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}) &= \sum_{i_T=1}^{N} P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}, X_T = s_{i_T}) \\
&= \sum_{i_T=1}^{N} \gamma_T(i_T)
\end{aligned}$$

If we examine the computation involved in the calculation of $\gamma_n(i_n)$, we see that it requires on the order of $N^2 \cdot (T+1)$ calculations, rather than $(T+1) \cdot N^{(T+1)}$ as required by the direct calculation. Hence, the forward probability calculation is more efficient than the direct calculation.

In similar, we have another method for the evaluation problem. It is called backward procedure.

Definition 2.2.2. ***(The Backward Procedure)***

We consider a backward variable $\beta_n(i_n)$ defined as,

$$\beta_n(i_n) = P_\lambda(\mathbf{Y^T_{n+1}} = \mathbf{r^T_{n+1}}|X_n = s_{i_n}) \quad 0 \le n \le T-1,\ 1 \le i_n \le N.$$

That is the probability of the partial observation sequence from time $n+1$ to the end, given the hidden state is s_{i_n} at time n. Again we can solve for $\beta_n(i_n)$ inductively, as follows:

1. *Initialization:*

   To make this procedure work for $(n = T-1)$, we arbitrarily define $\beta_T(i_T)$ to be 1 in the initialization step (1).

$$\beta_T(i_T) = 1.$$

2. *Induction $(n = 0, 1, \ldots, T-1)$:*

$$\begin{aligned}
&\beta_n(i_n)\\
= \;& P_\lambda(\mathbf{Y^T_{n+1}} = \mathbf{r^T_{n+1}}|X_n = s_{i_n})\\
= \;& \sum_{i_{n+1}=1}^{N} P_\lambda(\mathbf{Y^T_{n+1}} = \mathbf{r^T_{n+1}}, X_{n+1} = s_{i_{n+1}}|X_n = s_{i_n})\\
= \;& \sum_{i_{n+1}=1}^{N} P_\lambda(\mathbf{Y^T_{n+1}} = \mathbf{r^T_{n+1}}|X_{n+1} = s_{i_{n+1}}, X_n = s_{i_n}) \cdot P_\lambda(X_{n+1} = s_{i_{n+1}}|X_n = s_{i_n})\\
= \;& \sum_{i_{n+1}=1}^{N} P_\lambda(\mathbf{Y^T_{n+1}} = \mathbf{r^T_{n+1}}|X_{n+1} = s_{i_{n+1}}) \cdot P_\lambda(X_{n+1} = s_{i_{n+1}}|X_n = s_{i_n})\\
= \;& \sum_{i_{n+1}=1}^{N} \sum_{1 \le i_{n+2},\ldots,i_T \le N} P_\lambda(\mathbf{Y^T_{n+1}} = \mathbf{r^T_{n+1}}|\mathbf{X^T_{n+1}} = \mathbf{s^T_{n+1}}) \cdot P_\lambda(\mathbf{X^T_{n+2}} = \mathbf{s^T_{n+2}}|X_{n+1} = s_{i_{n+1}})\\
& \cdot P_\lambda(X_{n+1} = s_{i_{n+1}}|X_n = s_{i_n})
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i_{n+1}=1}^{N} P_\lambda(Y_{n+1}=r_{j_{n+1}}|X_{n+1}=s_{i_{n+1}})\cdot P_\lambda(X_{n+1}=s_{i_{n+1}}|X_n=s_{i_n})\\
&\quad\cdot[\sum_{1\le i_{n+2},\dots,i_T\le N} P_\lambda(\mathbf{Y^T_{n+2}}=\mathbf{r^T_{n+2}}|\mathbf{X^T_{n+1}}=\mathbf{s^T_{n+1}})P_\lambda(\mathbf{X^T_{n+2}}=\mathbf{s^T_{n+2}}|X_{n+1}=s_{i_{n+1}})]\\
&= \sum_{i_{n+1}=1}^{N} P_\lambda(Y_{n+1}=r_{j_{n+1}}|X_{n+1}=s_{i_{n+1}})\cdot P_\lambda(\mathbf{Y^T_{n+2}}=\mathbf{r^T_{n+2}}|X_{n+1}=s_{i_{n+1}})\\
&\quad\cdot P_\lambda(X_{n+1}=s_{i_{n+1}}|X_n=s_{i_n})\\
&= \sum_{i_{n+1}=1}^{N} p_{i_n i_{n+1}} b_{i_{n+1}}(j_{n+1})\beta_{n+1}(i_{n+1}).
\end{aligned}
$$

3. *Termination:*

$$P_\lambda(\mathbf{Y^T_0}=\mathbf{r^T_0})$$

$$
\begin{aligned}
&= \sum_{i_0=1}^{N} P_\lambda(\mathbf{Y^T_0}=\mathbf{r^T_0}, X_0=s_{i_0})\\
&= \sum_{i_0=1}^{N} P_\lambda(\mathbf{Y^T_0}=\mathbf{r^T_0}|X_0=s_{i_0})\cdot P_\lambda(X_0=s_{i_0})\\
&= \sum_{i_0=1}^{N} P_\lambda(\mathbf{Y^T_1}=\mathbf{r^T_1}|X_0=s_{i_0})\cdot P_\lambda(Y_0=r_{j_0}|X_0=s_{i_0})\cdot P_\lambda(X_0=s_{i_0})\\
&= \sum_{i_0=1}^{N} \beta_0(i_0)b_{i_0}(j_0)\alpha_{i_0}
\end{aligned}
$$

The backward procedure requires on the order of $N^2\cdot(T+1)$ calculations, as many as the forward procedure.

Summary of the evaluation problem.

We have introduced how to evaluate the probability that the observation sequence $\mathbf{Y^T_0}=\mathbf{r^T_0}$ is generated by using either the forward procedure or the backward procedure. They are more efficient than the method given in Lemma 2.2.1.

In fact, these two procedures are nothing but changing multiple sum, that is Lemma 2.2.1, to repeated sum. For example, in the forward procedure, we use the identity

$$\sum_{1\le i_0,\dots,i_T\le N} * = \sum_{i_T=1}^{N}\dots\sum_{i_0=1}^{N} *,$$

and for the backward procedure, we reverse the order of summation, that is

$$\sum_{1\le i_0,\ldots,i_T\le N} * = \sum_{i_0=1}^{N}\cdots\sum_{i_T=1}^{N} *.$$

From this point of view, it is obvious that other procedures are possible. For example, we can do the summation from the two ends to the middle at the same time.

2.The Decoding problem

Given a model λ and a sequence of observations $\mathbf{Y_0^T} = \mathbf{r_0^T}$, what is the most likely state sequence in the model that produced the observation? That is, we want to find a hidden state sequence $\mathbf{X_0^T} = \mathbf{s_0^T}$, to maximize the probability, $P_\lambda(\mathbf{X_0^T} = \mathbf{s_0^T}|\mathbf{Y_0^T} = \mathbf{r_0^T})$, for any possible sequences $\mathbf{s_0^T}$. This is equivalent to maximize $P_\lambda(\mathbf{X_0^T} = \mathbf{s_0^T}, \mathbf{Y_0^T} = \mathbf{r_0^T})$, because the probability of $\mathbf{r_0^T}$ given a model λ, $P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})$ is fixed. A technique for finding this state sequence exists, based on dynamic programming methods, and is called the Viterbi algorithm [26].

Definition 2.2.3. ***(Viterbi algorithm)***

Fixing s_{i_n}, we consider a variable $\rho_n(i_n)$ defined as

$$\rho_n(i_n) = \max_{1\le i_0,i_1,\ldots,i_{n-1}\le N} P_\lambda(\mathbf{X_0^{n-1}} = \mathbf{s_0^{n-1}}, X_n = s_{i_n}, \mathbf{Y_0^n} = \mathbf{r_0^n}).$$

Hence, $\rho_n(i_n)$ is the highest probability along a path, which accounts for the first n observations and ends in state s_{i_n} at time n.

By induction we have,

$$\begin{aligned}
&\rho_{n+1}(i_{n+1})\\
&= \max_{1\le i_0,i_1,\ldots,i_n\le N} P_\lambda\left(\mathbf{X_0}^{\mathbf{n}} = \mathbf{s_0}^{\mathbf{n}}, X_{n+1} = s_{i_{n+1}}, \mathbf{Y_0}^{\mathbf{n+1}} = \mathbf{r_0}^{\mathbf{n+1}}\right)\\
&= \max_{1\le i_0,i_1,\ldots,i_n\le N} P_\lambda\left(\mathbf{Y_0}^{\mathbf{n+1}} = \mathbf{r_0}^{\mathbf{n+1}} \mid \mathbf{X_0}^{\mathbf{n+1}} = \mathbf{s_0}^{\mathbf{n+1}}\right)\cdot P_\lambda\left(\mathbf{X_0}^{\mathbf{n+1}} = \mathbf{s_0}^{\mathbf{n+1}}\right)\\
&= \max_{1\le i_0,i_1,\ldots,i_n\le N} P_\lambda\left(Y_{n+1} = r_{j_{n+1}} \mid X_{n+1} = s_{i_{n+1}}\right)\cdot P_\lambda\left(\mathbf{Y_0}^{\mathbf{n}} = \mathbf{r_0}^{\mathbf{n}} \mid \mathbf{X_0}^{\mathbf{n}} = \mathbf{s_0}^{\mathbf{n}}\right)\\
&\quad \cdot P_\lambda\left(X_{n+1} = s_{i_{n+1}} \mid \mathbf{X_0}^{\mathbf{n}} = \mathbf{s_0^n}\right)\cdot P_\lambda\left(\mathbf{X_0}^{\mathbf{n}} = \mathbf{s_0}^{\mathbf{n}}\right)\\
&= \max_{1\le i_0,i_1,\ldots,i_n\le N} P_\lambda\left(Y_{n+1} = r_{j_{n+1}} \mid X_{n+1} = s_{i_{n+1}}\right)\cdot P_\lambda\left(\mathbf{Y_0}^{\mathbf{n}} = \mathbf{r_0}^{\mathbf{n}}, \mathbf{X_0}^{\mathbf{n}} = \mathbf{s_0}^{\mathbf{n}}\right)\\
&\quad \cdot P_\lambda\left(X_{n+1} = s_{i_{n+1}} \mid X_n = s_{i_n}\right)\\
&= \left[\max_{1\le i_n\le N}\max_{1\le i_0,\ldots,i_{n-1}\le N} P_\lambda\left(\mathbf{Y_0^n} = \mathbf{r_0^n}, \mathbf{X_0}^{\mathbf{n}} = \mathbf{s_0}^{\mathbf{n}}\right)\cdot P_\lambda\left(X_{n+1} = s_{i_{n+1}} \mid X_n = s_{i_n}\right)\right]\\
&\quad \cdot P_\lambda\left(Y_{n+1} = r_{j_{n+1}} \mid X_{n+1} = s_{i_{n+1}}\right)\\
&= \left[\max_{1\le i_n\le N} \rho_n\left(i_n\right) p_{i_n i_{n+1}}\right] b_{i_{n+1}}\left(j_{n+1}\right)
\end{aligned}$$

To find the hidden state sequence, we need to keep track of the argument which maximize $\rho_n(i_n)$, for every n and i_n, and they will be noted by an array $\phi_n(i_n)$.

1. *Initialization:*

$$\rho_0(i_0) = \alpha_{i_0} b_{i_0}(j_0), \quad 1 \le i_0 \le N.$$

2. *Induction $(n = 1, \ldots, T)$:*

$$\begin{aligned} \rho_n(i_n) &= \max_{1 \le i_{n-1} \le N} [\rho_{n-1}(i_{n-1}) p_{i_{n-1} i_n}] b_{i_n}(j_n), \\ \phi_n(i_n) &= \arg \max_{1 \le i_{n-1} \le N} [\rho_{n-1}(i_{n-1}) p_{i_{n-1} i_n}]. \end{aligned}$$

3. *Termination:*

$$\begin{aligned} \rho^* &= \max_{1 \le i_T \le N} [\rho_T(i_T)], \\ \phi^* &= \arg \max_{1 \le i_T \le N} [\rho_T(i_T)]. \end{aligned}$$

Finally, we will have the state sequence $i_T = \phi^*$ and $i_n = \phi_{n+1}(i_{n+1})$, $n = T-1, T-2, \ldots, 0$.

The Viterbi algorithm is very similar as the forward procedure that we have introduced in the above section. In the forward procedure, we define $\gamma_n(i_n)$, while here we use $\rho_n(i_n)$. The only difference between them is that we change summation to maximum. But the general ideas are same. We change multiple summation to repeated summation and multiple maximum to repeated maximum.

From this point of view, it is very natural to think how to use the idea that we have used in the backward procedure to solve the decoding problem.

3.The Learning problem

Given a model λ and a sequence of observations $\mathbf{Y_0^T} = \mathbf{r_0^T}$, how should we adjust the model parameters $(\mathbf{A}, \mathbf{B}, \alpha)$ in order to maximize $P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})$? We face an optimization problem with restrictions. This probability $P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})$ is a function of the variables $\alpha_i, p_{ij}, b_j(k)$. Also we may view the probability $P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})$ as the likelihood function of λ, considered as a function of λ for fixed $\mathbf{r_0^T}$. Thus, for each $\mathbf{r_0^T}$, $P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})$ gives the probability of observing $\mathbf{r_0^T}$. We use the method of maximum likelihood, try to find that the value of λ, that is "most likely" to have produced the sequence $\mathbf{r_0^T}$. Lemma 2.2.1.

The problem we need to solve is:

$$\max_\lambda P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}) = \max_\lambda \sum_{1 \le i_0, i_1, \ldots, i_T \le N} \prod_{n=0}^{T} p_{i_{n-1} i_n} b_{i_n}(j_n),$$

To solve this problem, we need to use some results that **Leonard E. Baum and George R** proved in 1968 [6]. We introduce an algorithm to solve our problem. That is, **the Baum-Welch algorithm:** The Baum-Welch algorithm finds a local maximum for $\lambda^* = \arg\max_\lambda P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})$ (i.e. the HMM parameters λ that maximize the probability of observation).

We can now calculate the temporary variables, according to Bayes theorem. From the definition of the forward and backward variables, we can write

$$\omega_n(i_n) = P_\lambda(X_n = s_{i_n} \mid \mathbf{Y_0^T} = \mathbf{r_0^T}) = \frac{P_\lambda(X_n = s_{i_n}, \mathbf{Y_0^T} = \mathbf{r_0^T})}{P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})} = \frac{\gamma_n(i_n)\beta_n(i_n)}{\sum_{i_{n+1}}^N \gamma_{n+1}(i_{n+1})\beta_{n+1}(i_{n+1})}.$$

which is the probability of being in state s_{i_n} at time n, given the observation sequence and the model λ.

We define $\xi_n(i_n, i_{n+1})$ the probability of being in state s_{i_n} at time n and state $s_{i_{n+1}}$ at time n+1, given the model λ and the observation sequence

$$\begin{aligned}\xi_n(i_n, i_{n+1}) &= P_\lambda(X_n = s_{i_n}, X_{n+1} = s_{i_{n+1}} | \mathbf{Y_0^T} = \mathbf{r_0^T}) \\ &= \frac{P_\lambda(X_n = s_{i_n}, X_{n+1} = s_{i_{n+1}}, \mathbf{Y_0^T} = \mathbf{r_0^T})}{P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})} \\ &= \frac{\gamma_n(i_n) p_{i_n i_{n+1}} b_{i_{n+1}}(j_{n+1}) \beta_{n+1}(i_{n+1})}{\sum_{i_n=1}^N \sum_{i_{n+1}=1}^N \gamma_n(i_n) p_{i_n i_{n+1}} b_{i_{n+1}}(j_{n+1}) \beta_{n+1}(i_{n+1})}.\end{aligned}$$

The denominators of $\omega_n(i_n)$ and $\xi_n(i_n, i_{n+1})$ are the same, they represent the probability of making the observation Y given λ.

The parameters of the hidden Markov model λ can now be updated:

$$\begin{aligned}
\overline{\alpha_i} &= expected\, frequency\,(number\, of\, times)\, in\, state\, s_i\, at\, time\, 0 \\
&= \omega_0(i). \\
\overline{p_{ij}} &= \frac{expected\, number\, of\, transitions\, from\, s_i\, to\, s_j}{expected\, number\, of\, transitions\, from\, s_i} \\
&= \frac{\displaystyle\sum_{n=0}^{T-1} \xi_n(i,j)}{\displaystyle\sum_{n=0}^{T-1} \omega_n(i)}. \\
\overline{b_j(k)} &= \frac{expected\, number\, of\, times\, in\, state\, s_j\, and\, observation\, is\, r_k}{expected\, number\, of\, times\, in\, state\, s_j} \\
&= \frac{\displaystyle\sum_{n=0}^{T-1} \mathbb{I}_{\{Y_n=r_k\}}\omega_n(j)}{\displaystyle\sum_{n=0}^{T-1} \omega_n(j)}.
\end{aligned}$$

Therefore, we obtain a new model $\bar{\lambda} = (\bar{\mathbf{A}}, \overline{\mathbf{B}}, \bar{\alpha})$. Based on the above procedure, if we iteratively use $\bar{\lambda}$ in place of λ and repeat the calculation, we can improve the probability of $\mathbf{Y_0^T} = \mathbf{r_0^T}$, until some limiting point is reached.

2.3 Hidden semi-Markov model

2.3.1 General structure of HSMM

The HSMMs capacity extends beyond that of the HMM. It allows for every hidden state to be a semi-Markov chain while also introducing the concept of state duration. This means that, unlike in HMM where a state can emit one observation per state, a state in HSMM can emit a sequence of characters. The length of the observation sequence for each state is determined by the duration variable (sojourn times) of each state. Consequently, in addition to the standard notation of a HMM, a state duration variable is added for the HSMM. This is an integer variable and takes the value from the set $d = \{1, 2, \ldots, D\}$, where D is the maximum duration allowed for a single state.

Below is a figure depicting the general HSMM structure. The initial state and its duration are selected according to the initial transition probabilities. In this case, the first state produces two observations hence the duration equals two and transitions into the second state. The second state then produces an observation sequence length of four. This can be seen for the remaining states till time (T).

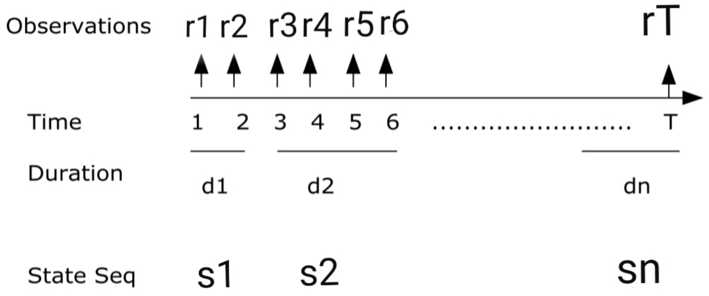


Figure 2.3: General model of HSMM

Before giving any formal definition of the hidden semi-Markov model, let us first see a concrete application in genetics

Example 2. ***(CpG islands in a DNA sequence).***

Consider a DNA sequence, that is, a sequence of the four nucleotides A, C, G, and T, i.e., an element of the space $\{A, C, G, T\}^{\mathbb{N}}$,

$$\{T\,A\,G\,T\,G\,G\,A\,A\,C\,G\,A\,C\,C\,G\,G\,A\,T\,C\,C\ldots\}.$$

It is known that the presence of the pairs C-G is relatively rare in the genome. Nevertheless, there are some regions within a DNA sequence where the frequency of C-G pairs, as well as the frequency of nucleotides C and G themselves, is more important. It has been proved that these regions, called CpG islands, play a key role in the coding mechanism, so finding them is of great importance in genetic research. Several mathematical models have been proposed for detecting CpG islands [14]. We will present in the sequel the use of the hidden Markov model for detection of CpG islands and we will also see why we think that it is more natural to use a hidden semi-Markov model instead. Suppose that the DNA sequence is modeled by a sequence of conditionally independent random variables Y, with state space $D = \{A, C, G, T\}$. *Suppose also that the possible presence of a CpG island is modeled by a Markov chain Z with state space* $E = \{0, 1\}$. *Having* $(y_0, \ldots, y_M)$ *a truncated sample path of Y, we set* $Z_n = 1$ *if* y_n *is a nucleotide inside a CpG island and* $Z_n = 0$ *otherwise :*

- $Y: \underbrace{TAGTGGAATG}\underbrace{CGACG}\ldots -$ *DNA sequence*
- $Z: 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ldots -$ *CpG islands indicators.*

Suppose that $(Z_n)_{n \in \mathbb{N}}$ *is a Markov chain and that the observed nucleotides* Y_n *are generated according to the corresponding* Z_n*. This is a typical example of a hidden Markov model. From a practical point of view, the main drawback of this type of approach is that we suppose that the length of windows of 0s and 1s follows geometric distributions since we impose a Markovian evolution on process Z. For this reason it is more natural to let Z be a semi-Markov chain, allowing a more realistic behavior of the model, as the length of windows of 0s and 1s can follow any probability distribution on* $\mathbb{N}$*, instead of a geometric one in the Markov case. In this way, we obtain what is called a hidden semi-Markov model. Obviously, the HMM is a particular case of the HSMM.*

Example 3. ***(Hidden Markov chains for detecting an unfair die)****.*

Consider two dice, a fair one and an unfair one. When rolling the unfair die, there is a $1/2$ *probability of getting a 6 and a* $1/10$ *probability of getting 1,2,3,4, or 5. After rolling the fair die, the probability that the next game will be done with the unfair die is* $1/10$*. On the other hand, after rolling the unfair die, the probability that the next game will be done with the fair die is* $1/2$*.*

Let $Z_0, Z_1, \ldots$ *be the random variable sequence of successively used dice, with value 0 for the fair die and 1 for unfair one. Consider also* $Y_0, Y_1, \ldots$ *the random variable sequence, with values in* $\{1, 2, 3, 4, 5, 6\}$ *representing the successive values of the rolled dice. In practical terms, only sequence Y is observed, whereas chain Z is "hidden" (unobserved, unknown). The couple* (Z, Y) *is a hidden Markov chain, that is, Z is an unobserved ergodic Markov chain and Y is a sequence of conditional independent random variables, in the sense that the distribution of* Y_n *depends only on* $Z_n, n \in \mathbb{N}$*.*
Let us compute:

1. *The probability* $\mathbb{P}(Y_n = i | Z_0 = 1)$, $1 \leq i \leq 6$,
2. *The limit* $\lim_{n\to\infty} \mathbb{P}(Y_n = i | Z_0 = 1)$,$1 \leq i \leq 6$.

Set $E = \{0, 1\}$ *for the state space of the Markov chain Z and note that its associated transition matrix is*

$$\mathbf{P} = \begin{pmatrix} 9/10 & 1/10 \\ 1/2 & 1/2 \end{pmatrix},$$

the conditional distributions of Y_n, given the state of $Z_n, n \in \mathbb{N}$, are as follows:

$$\begin{aligned}
\mathbb{P}(Y_n = i|Z_n = 1) &= 1/10, \, for\, all\, i = 1, \dots, 5, \\
\mathbb{P}(Y_n = 6|Z_n = 1) &= 1/2, \\
\mathbb{P}(Y_n = i|Z_n = 0) &= 1/6, \, for\, all\, i = 1, \dots, 6.
\end{aligned}$$

Let us now compute the probabilities of interest.

1. *We have*

$$\begin{aligned}
\mathbb{P}(Y_n = i|Z_0 = 1) &= \sum_{l \in E} \mathbb{P}(Y_n = i, Z_n = l|Z_0 = 1) \\
&= \sum_{l \in E} \mathbb{P}(Y_n = i|Z_n = l, Z_0 = 1)\mathbb{P}(Z_n = l|Z_0 = 1) \\
&= \sum_{l \in E} \mathbb{P}(Y_n = i|Z_n = l)p_{1l}^n,
\end{aligned}$$

where p_{1l}^n is the element $(1, l)$ of $\mathbf{p}^n$, the n-fold matrix product of $\mathbf{p}$. Using the previous computations, given a state i and a positive integer n, one can immediately obtain the values of $\mathbb{P}(Y_n = i|Z_0 = 1)$.

2. *To obtain the limit $\lim_{n\to\infty} \mathbb{P}(Y_n = i|Z_0 = 1)$,$1 \leq i \leq 6$, we start with the relation obtained above,*

$$\mathbb{P}(Y_n = i|Z_0 = 1) = \sum_{l \in E} \mathbb{P}(Y_n = i|Z_n = l)p_{1l}^n.$$

First, note that the probabilities $\mathbb{P}(Y_n = i|Z_n = l)$ do not depend on $n \in \mathbb{N}$, so the limit as n tends to infinity concerns only p_{1l}^n. Second, from Proposition 1.2.3 we know that

$$\lim_{n\to\infty} p_{1l}^n = \nu(l)$$

where $\nu = (\nu(0)\, \nu(1))$ is the stationary (invariant) distribution of the Markov chain $(Z_n)_{n\in\mathbb{N}}$. We compute the stationary distribution ν by solving the system $\nu\, \mathbf{p} = \nu$, with the additional condition $\nu(0) + \nu(1) = 1$, and we get $\nu(0) = 5/6$, $\nu(1) = 1/6$. Consequently, we obtain

$$\begin{aligned}
&\lim_{n\to\infty} \mathbb{P}(Y_n = i|Z_0 = 1) \\
= \; &\mathbb{P}(Y_n = i|Z_n = 0)\nu(0) + \mathbb{P}(Y_n = i|Z_n = 1)\nu(1)
\end{aligned}$$

Finally, for $i = 1, \dots, 5$, we get

$$\lim_{n\to\infty} \mathbb{P}(Y_n = i|Z_0 = 1) = \frac{1}{6} \times \frac{5}{6} + \frac{1}{10} \times \frac{1}{6} = \frac{7}{45},$$

whereas for $i = 6$ the limit is

$$\lim_{n\to\infty} \mathbb{P}(Y_n = 6|Z_0 = 1) = \frac{1}{6} \times \frac{5}{6} + \frac{1}{2} \times \frac{1}{6} = \frac{2}{9}$$

.

Let us now formally define a hidden semi-Markov model. We will take into account two different types of such models, the so-called hidden SM-M0 model and the hidden SM-Mk, $k \geq 1$ model. Let $Z = (Z_n)_{n\in\mathbb{N}}$ be a semi-Markov chain with finite state space $\mathrm{E} = \{1, \ldots, s\}$ and $Y = (Y_N)_{n\in\mathbb{N}}$ be a stationary sequence of random variables with finite state space $A = \{1, \ldots, d\}$, i.e., for any $n \in \mathbb{N}$, $a \in A$, and $i \in \mathrm{E}$ we have that $\mathbb{P}(Y_n = a|Z_n = i)$ is independent of n.

Before giving the definitions, let us introduce some notation.

Notation. Let $l, k \in \mathbb{N}$ be two nonnegative integers such that $l \leq k$, and let $a_l, \ldots, a_k \in A$. We will denote by $\mathbf{Y_l^k}$ the vector $\mathbf{Y_l^k} = (Y_l, \ldots, Y_k)$ and we will write $\{\mathbf{Y_l^k} = a_l^k\}$ for the event $\{Y_l = a_l, \ldots, Y_k = a_k\}$. When all these states represent the same state, say $a \in A$, we simply denote by $\{\mathbf{Y_l^k} = a\}$ the event $\{Y_l = a, \ldots, Y_k = a\}$. We also denote by $\{\mathbf{Y_l^k} = \cdot\}$ the event $\{\mathbf{Y_l^k} = \cdot\} = \{Y_l = \cdot, \ldots, Y_k = \cdot\}$. We gave all this notation in terms of chain Y, but it can be obviously used for chain Z.

Example 4. ***(Hidden semi-Markov chains for detecting an unfair die.)*** *Let us consider the problem of an unfair die detection presented in Example 3 and see how we can propose a hidden semi-Markov modeling instead of a hidden Markov one.*

As before, we have two dice, an unfair one and a fair one. When rolling the unfair die, there is a 1/2 probability of getting a 6 and a 1/10 probability of getting 1, 2, 3, 4, or 5. After rolling the fair die n times, the probability that the next roll will be done with the unfair die is $f(n)$, where $f = (f(n))_{n\in\mathbb{N}^*}$. is a distribution on $\mathbb{N}^*$. On the other hand, after rolling the unfair die n times, the probability that the next roll will be done with the fair die is $g(n)$, where $g = (g(n))_{n\in\mathbb{N}^*}$. is a distribution on $\mathbb{N}^*$.

Let $Z_0, Z_1, \ldots$ be the random sequence of successively used dice, with value 0 for the fair die and 1 for the fake die. Consider also $Y_0, Y_1, \ldots$ the random variable sequence, with values in $\{1, 2, 3, 4, 5, 6\}$ representing the successive values of the rolled dice. The couple $(Z, Y) = (Z_n, Y_n)_{n\in\mathbb{N}}$ is a hidden semi- Markov chain of type SM-M0.

Definition 2.3.1. ***(Hidden semi-Markov chain of type SM-M0)***

1. *Let $Y = (Y_n)_{n\in\mathbb{N}}$ be conditionally independent random variables, given a sample path of the SMC Z, i.e., for all $a \in A$, $i \in \mathrm{E}$, $n \in \mathbb{N}^*$, the following relation holds true:*

$$\mathbb{P}(Y_n = a|\mathbf{Y_0^{n-1}} = \cdot, Z_n = i, \mathbf{Z_0^{n-1}} = \cdot) = \mathbb{P}(Y_n = a|Z_n = i). \tag{2.4}$$

   The chain $(Z, Y) = (Z_n, Y_n)_{n\in\mathbb{N}}$ is called a hidden semi-Markov chain of type SM-M0, where the index 0 stands for the order of Y regarded as a conditional Markov chain.

2. *For (Z,Y) a hidden semi-Markov chain of type SM-M0, let us define $R = (R_{i,a}, i \in \mathrm{E}, a \in A) \in \mathcal{M}_{\mathrm{E}\times A}$ as the conditional distribution of chain Y*

$$R_{i,a} = \mathbb{P}(Y_n = a|Z_n = i), \tag{2.5}$$

   called the emission probability matrix.

Definition 2.3.2. ***(Hidden semi-Markov chain of type SM-Mk)***

1. *Let $Y = (Y_n)_{n\in\mathbb{N}}$ be a homogeneous Markov chain of order k, $k \geq 1$, conditioned on the SMC Z, i.e., for all $a_0, \ldots, a_k \in A, i \in \mathrm{E}, n \in \mathbb{N}^*$, the following relation holds true:*

$$\begin{aligned}\mathbb{P}(Y_{n+1} = a_k|\mathbf{Y^n_{n-k+1}} = \mathbf{a_0^{k-1}}, \mathbf{Y_0^{n-k}} = \cdot, Z_{n+1} = i, \mathbf{Z_0^n} = \cdot)& \\ = \mathbb{P}(Y_{n+1} = a_k|\mathbf{Y^n_{n-k+1}} = \mathbf{a_0^{k-1}}, Z_{n+1} = i)&\end{aligned} \tag{2.6}$$

   The chain $(Z, Y) = (Z_n, Y_n)_{n\in\mathbb{N}}$ is called a hidden semi-Markov chain of type SM-Mk, where the index k stands for the order of the conditional Markov chain Y.

2. *For (Z, Y) a hidden semi-Markov chain of type SM-Mk, let us define $R = (R_{i,a_0,\ldots,a_k}, i \in \mathrm{E}, a_0, \ldots, a_k \in A) \in \mathcal{M}_{\mathrm{E}\times A\times\ldots\times A}$ as the transition matrix of the conditional Markov chain Y*

$$R_{i,a_0,\ldots,a_k} = \mathbb{P}(Y_{n+1} = a_k|\mathbf{Y^n_{n-k+1}} = \mathbf{a_0^{k-1}}, Z_{n+1} = i), \tag{2.7}$$

   called the emission probability matrix of the conditional Markov chain Y.

2.3.2 Estimation of a Hidden Semi-Markov Model

Let $(Z, Y) = (Z_n, Y_n)_{n\in\mathbb{N}}$ be a hidden SM-M0 chain with finite state space $\mathrm{E} \times A$. We suppose that the semi-Markov chain Z is not directly observed and that the observations are described by the sequence of conditionally independent random variables $Y = (Y_n)_{n\in\mathbb{N}}$. Starting from a sample path $y = \mathbf{y_0^M} = (y_0, \ldots, y_M)$ of observations, we want to estimate the characteristics of the underlying semi-Markov chain, as well as the conditional distribution of $Y = (Y_n)_{n\in\mathbb{N}}$. All the results are obtained under Assumption **A1** of Chapter 1, that the SMC $(Z_n)_{n\in\mathbb{N}}$ is irreducible.

2.3.3 Consistency of Maximum-Likelihood Estimator

Let $U = (U_n)_{n\in\mathbb{N}}$ be the backward-recurrence times of the semi-Markov chain $(Z_n)_{n\in\mathbb{N}}$, that is,

$$U_n = n - S_{N(n)} \tag{2.8}$$

One can check that the chain $(Z, U) = (Z_n, U_n)_{n\in\mathbb{N}}$ is a Markov chain with state space $\mathrm{E} \times \mathbb{N}$. Let us denote by $\tilde{\mathbf{p}} = (p_{(i,t_1)(j,t_2)})_{i,j\in\mathrm{E},t_1,t_2\in\mathbb{N}}$ its transition matrix. We can easily prove the following result, which gives the transition matrix $\tilde{\mathbf{p}}$ in terms of the semi-Markov kernel $\mathbf{q}$.

Proposition 2.1. *For all $i, j \in \mathrm{E}$, $t_1, t_2 \in \mathbb{N}$, the transition probabilities of the Markov chain (Z, U) are given by:*

$$p_{(i,t_1)(j,t_2)} = \begin{cases} q_{ij}(t_1+1)/\overline{H_i}(t_1), & if\, i \neq j\, and\, t_2 = 0, \\ \overline{H_i}(t_1+1)/\overline{H_i}(t_1), & if\, i = j\, and\, t_2 - t_1 = 1, \\ 0, & otherwise. \end{cases} \tag{2.9}$$

where $\overline{H_i}(\cdot)$ is the survival function of sojourn time in state i (final Equation in definition 1.3.6).

Proposition 2.2. ***(Stationary distribution of the MC (Z,U)).***

Consider an aperiodic MRC (J_n, S_n) that satisfies Assumptions A1 and A2. Then the stationary probability distribution $\tilde{\pi} = (\pi_{i,u})_{i\in\mathrm{E},u\in\mathbb{N}}$ of the Markov chain (Z,U) is given by

$$\pi_{i,u} = \frac{1 - H_i(u)}{\mu_{ii}}. \tag{2.10}$$

We shall consider that the conditional distributions of sojourn times, $f_{ij}(\cdot)$, $i, j \in \mathrm{E}, i \neq j$, have the same bounded support, $supp\, f_{ij}(\cdot) = \mathrm{D} = \{1, \ldots, \tilde{n}\}$ for all $i, j \in \mathrm{E}, i \neq j$.

We suppose that the Markov chain $(Z_n, U_n)_{n\in\mathbb{N}}$ has the finite state space $\mathrm{E} \times \mathrm{D}$ and the transition matrix $\tilde{\mathbf{p}} = (p_{(i,t_1)(j,t_2)})_{i,j\in\mathrm{E},t_1,t_2\in\mathrm{D}}$. All the work in the rest of this chapter will be done under the assumption:

A4 The conditional sojourn time distributions have finite support D.

Taking into account the conditional independence, Relation 2.4, for all $a \in \mathrm{A}$, $j \in \mathrm{E}$, and $t \in \mathrm{D}$, we have

$$R_{i,a} = \mathbb{P}(Y_n = a | Z_n = i) = \mathbb{P}(Y_n = a | Z_n = i, U_n = t). \tag{2.11}$$

Consequently, starting from the initial hidden semi-Markov chain $(Z_n, Y_n)_{n \in \mathbb{N}}$, we have an associated hidden Markov chain $((Z, U), Y) = ((Z_n, U_n), Y_n)_{n \in \mathbb{N}}$, with $(Z_n, U_n)_{n \in \mathbb{N}}$ a Markov chain and $(Y_n)_{n \in \mathbb{N}}$ a sequence of conditionally independent random variables. This new hidden Markov model is defined by:

- The transition matrix $\tilde{\mathbf{p}} = (p_{(i,t_1)(j,t_2)})_{i,j \in \mathrm{E}, t_1, t_2 \in \mathrm{D}}$ of the Markov chain (Z, U), with $p_{(i,t_1)(j,t_2)}$ given by Equation 2.9,
- The conditional distribution $\mathbf{R}$ of the sequence Y, given by Equation 2.11,
- The initial distribution of the hidden Markov chain $((Z, U), Y)$.

We consider that the HSMM is stationary. Consequently, we will not take into account the initial distribution in the parameter space.

In order to obtain the parameter space of the hidden Markov model, note first that

- $q_{ij}(t_1 + 1) = 0$ for $t_1 + 1 > \tilde{n}$,
- $\overline{H_i}(t_1 + 1) = \sum_{k=t_1+2}^{\infty} \sum_{j \in \mathrm{E}} q_{ij}(k) = 0$ for $t_1 + 2 > \tilde{n}$.

Thus, for all $i, j \in \mathrm{E}, t_1, t_2 \in \mathbb{N}$, the transition probabilities of the Markov chain (Z, U) given in Proposition 2.9 can be written for our model as follows:

$$p_{(i,t_1)(j,t_2)} = \begin{cases} q_{ij}(t_1 + 1)/\overline{H_i}(t_1), & if\, i \neq j, t_2 = 0, and\, 0 \leq t_1 \leq \tilde{n} - 1, \\ \overline{H_i}(t_1 + 1)/\overline{H_i}(t_1), & if\, i = j\, t_2 - t_1 = 1, and\, 0 \leq t_1 \leq \tilde{n} - 2, \\ 0, & \text{otherwise}. \end{cases} \tag{2.12}$$

We denote the parameter θ by:

$$\theta = (\theta_1, \theta_2) = (\theta_1, \ldots, \theta_b) = ((p_{(i,t_1)(j,t_2)})_{i,j,t_1,t_2}, (R_{ia})_{i,a}).$$

Note that in the description of the parameter θ in terms of $p_{(i,t_1)(j,t_2)}$ and R_{ia} we consider only the non identically zero parameters, and all the dependent parameters have been removed, as described above. When we will need to consider also the dependent parameters $R_{i,d}, i \in \mathrm{E}$, (as in Theorem 2.5) we will denote the entire matrix of the conditional distribution of Y by $(R_{ia})_{i\in \mathrm{E}, a\in A}$ instead of $(R_{ia})_{i,a}$. Let us also denote by $\theta^{\mathbf{0}} = (\theta_1^0, \theta_2^0) = ((p^0_{(i,t_1)(j,t_2)})_{i,j,t_1,t_2}, (R^0_{ia})_{i,a})$ the true value of the parameter.

For $(Y_0, \dots, Y_M)$ a sample path of observations, the likelihood function for an observation of the hidden Markov chain $((Z,U),Y)$ is given by

$$p_\theta(Y_0^n) = \sum_{z_0^M, u_0^M} \pi_{z_0,u_0} \prod_{k=1}^{M} p_{(z_k-1,u_k-1)(z_k,u_k)} \prod_{k=0}^{M} R_{z_k,Y_k}. \tag{2.13}$$

As all the chains are assumed to be stationary, we consider that the time scale of all the processes is $\mathbb{Z}$ instead of $\mathbb{N}$, so we shall work with $(Z,U) = (Z_n, U_n)_{n\in\mathbb{Z}}$ and $Y = (Y_n)_{n\in\mathbb{Z}}$ instead of $(Z,U) = (Z_n,U_n)_{n\in\mathbb{N}}$ and $Y = (Y_n)_{n\in\mathbb{N}}$.

We have the consistency of the MLE of $\theta^{\mathbf{0}}_{\mathbf{1}} = (p^0_{(i,t_1)(j,t_2)})_{i,j,t_1,t_2}$ and $\theta^{\mathbf{0}}_{\mathbf{2}} = (R^0_{ia})_{i,a}$, denoted by $\widehat{\theta_1}(M) = (\hat{p}_{(i,t_1)(j,t_2)}(M))_{i,j,t_1,t_2}$ and $\widehat{\theta_{\mathbf{2}}}(M) = (\hat{R}_{ia}(M))_{i,a}$. Let us define this theorem,

Theorem 2.1. *[4] Under assumptions A1 and A4, given a sample of observations* $\mathbf{Y_0^M}$*, the maximum-likelihood estimator* $\hat{\theta}(M)$ *of* $\theta^{\mathbf{0}}$ *is consistent as* M *tends to infinity.*

The following two theorems use these results in order to prove the consistency of the maximum-likelihood estimators of the true value of the semi-Markov kernel $(q^0_{ij}(k))_{i,j\in\mathrm{E}, i\neq j, k\in\mathrm{D}}$ and of the true value of the transition matrix of the embedded Markov chain $(p^0_{i,j})_{i,j\in\mathrm{E}, i\neq j}$.

Theorem 2.2. *[4] Under assumptions A1 and A4, given a sample of observations* Y_0^M*, the maximum-likelihood estimator* $(\widehat{q_{ij}}(k,M))_{i,j\in\mathrm{E},i\neq j,k\in\mathrm{D}}$ *of* $(q^0_{ij}(k))_{i,j\in\mathrm{E},i\neq j,k\in\mathrm{D}}$ *is strongly consistent as* M *tends to infinity.*

Theorem 2.3. *[4] Under Assumptions* A1 *and* A4*, given a sample of observations* Y_0^M*, the maximum-likelihood estimator* $(\widehat{p}_{i,j}(M))_{i,j\in E,i\neq j}$ *of* $\left(p^0_{i,j}\right)_{i,j\in E,i\neq j}$ *is strongly consistent as* M *tends to infinity.*

2.3.4 Asymptotic Normality of Maximum-Likelihood Estimator

For $(Y_0, \dots, Y_n)$ a sample path of observations we denote by $\sigma_{Y_0^n}(\theta^0) = -\mathbb{E}_{\theta^\circ}\left(\left.\frac{\partial^2 \log p\left(Y_0^n\right)}{\partial\boldsymbol{\theta}_u \partial\theta_v}\right|_{\theta=\theta^\circ} \right)_{u,v}$ the Fisher information matrix computed in θ^0, where $p_{\boldsymbol{\theta}}\left(Y_0^n\right)$ is the associated likelihood function given in equation 2.13.

Let

$$\sigma\left(\theta^0\right) = \left(\sigma_{u,v}\left(\theta^0\right)\right)_{u,v} = -\mathbb{E}_{\theta^\circ}\left(\left.\frac{\partial^2 \log \mathbb{P}_\theta\left(Y_0 \mid Y_{-1}, Y_{-2}, \ldots\right)}{\partial\theta_u \partial\theta_v}\right|_{\theta=\theta^\circ}\right)_{u,v}$$

be the asymptotic Fisher information matrix computed in θ^0 ([5],[13]).

From Theorem 3 of Douc (2005) we know that $\sigma\left(\theta^0\right)$ is nonsingular if and only if there exists an integer $n \in \mathbb{N}$ such that $\sigma_{Y_0^n}\left(\theta^0\right)$ is nonsingular. Consequently, all our work will be done under the following assumption.

A5 There exists an integer $n \in \mathbb{N}$ such that the matrix $\sigma_{Y_0{}^n}\left(\theta^0\right)$ is nonsingular.

Theorem 2.4. *[4] Under Assumptions* $A1, A4$*, and* $A5$*, the random vector*

$$\sqrt{M}\left[\widehat{\theta}(M) - \theta^0\right] = \sqrt{M}\left[\left(\left(\widehat{p}_{(i,t_1)(j,t_2)}(M)\right)_{i,j,t_1,t_2}, \left(\widehat{R}_{ia}(M)\right)_{i,a}\right)\right.$$
$$\left. - \left(\left(p^0_{(i,t_1)(j,t_2)}\right)_{i,j,t_1,t_2}, \left(R^0_{ia}\right)_{i,a}\right)\right]$$

is asymptotically normal, as $M \to \infty$*, with zero mean and covariance matrix* $\sigma\left(\theta^0\right)^{-1}$

From this theorem we immediately obtain the asymptotic normality of the conditioned transition matrix R *of chain* Y*.*

Theorem 2.5. *[4] Under Assumptions* $A1, A4$*, and* $A5$*, the random vector*

$$\sqrt{M}\left[\left(\widehat{R}_{ia}(M)\right)_{i\in E, a\in A} - \left(R^0_{ia}\right)_{i\in E, a\in A}\right]$$

is asymptotically normal, as $M \to \infty$*.*

The following result concerns the asymptotic normality of the semi-Markov kernel estimator.

Theorem 2.6. *[4] Under Assumptions A1, A4, and A5, the random vector*

$$\sqrt{M}\left[\left(\widehat{q}_{ij}(k, M)\right)_{i,j\in E, i\neq j, k\in D} - \left(q^0_{ij}(k)\right)_{i,j\in E, i\neq j, k\in D}\right] \tag{2.14}$$

is asymptotically normal, as $M \to \infty$*.*

Chapter 3

R packages for analyzing SMM, HMM, HSMM

3.1 Introduction

R is a programming language and software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. R is an integrated suite of software facilities for data manipulation, calculation and graphical display.

The R language was designed in the 1980s and has been in widespread use in the statistical community since. The RStudio is an integrated development environment for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management. In this chapter will depends on this programming language, and we will present some R packages which can be used in our work, we talk about (**SMM,HMM,hsmm**) packages, respectively used for simulation and estimation of the semi Markov models, hidden Markov, and Hidden semi-Markov models.

3.2 Package SMM

In this section we will present the R package **(SMM)** which is developed by Vlad and al. This package performs parametric and non-parametric estimation and simulation for a Markovien process and multi-state discrete-time semi-Markov processes. For the parametric estimation, several discrete distributions are considered for the sojourn times (Uniform, Geometric, Poisson, Discrete Weibull and Negative Binomial). The non-parametric estimation concerns the sojourn time distributions, where no assumptions are done on the shape of distributions. Moreover, the estimation can be done on the basis of one or several sample paths, with or without censoring at the beginning or at the end of the sample paths. Estimation and simulation of discrete-time k-th order Markov chains are also considered.

Remark 3.2.1. ○ *All along this section we assume that the MRC or SMC are homogeneous with respect to the time.*

The **SMM** R package is mainly devoted to the simulation and estimation of discrete-time semi-Markov models in different cases by the two following functions:

- **simulSM :** for the simulation of sequences from a semi-Markov model.
- **estimSM :** for the estimation of model parameters.

The **SMM** R package is also devoted to the simulation and estimation of discrete-time Markov models by the two following functions :

- **simulMk :** for the simulation of sequences from a kth order Markov model.
- **estimMk :** for the estimation of the parameters of the model.

3.2.1 Simulation of semi-Markov models

Parametric simulation: according to classical distributions

Parameters : This simulation is carried out by the function **simulSM()**. The different parameters of the function are :

- **E :** Vector of state space of length S.
- **NbSeq :** Number of simulated sequences.
- **lengthSeq :** Vector containing the lengths of each simulated sequence.
- **init :** Vector of initial distribution of length S.
- **Ptrans :** Matrix of transition probabilities of the embedded Markov chain $J = (J_m)_m$ of size $S \times S$.
- **distr :** Sojourn time distributions :

  - is a matrix of distributions of size $S \times S$.

where the distributions to be used can be one of "uniform", "geom", "pois", "weibull" or "nbinom".

- **param :** Parameters of sojourn time distributions :

  - is an array of parameters of size $S \times S \times 2$

- **File.out :** Name of fasta file for saving the sequences, if **File.out** = **NULL**, no file is created. A **fasta** file is a simple text file containing sequences only described by one description line beginning by a " > ".

Example 5. *The R commands below generate a sequences of size 1000, with the finite state space $E = \{a, c, g, t\}$, where the sojourn times depend on the current state and on the next state.*

```
# state space
E = c("a","c","g","t")
S = length(E)
# sequence sizes
lengthSeq = c(1000)
# creation of the initial distribution
vect.init = c(1/4,1/4,1/4,1/4)
# creation of transition matrix
Pij = matrix(c(0,0.2,0.3,0.4,0.2,0,0.5,0.2,0.5,0.3,0,0.4,0.3,0.5,0.2,0),
ncol=4)
# creation of the distribution matrix
distr.matrix = matrix(c("dweibull", "pois", "geom", "nbinom", "geom", "nbinom", "pois", "dweibull","pois", "pois", "dweibull", "geom", "pois","geom", "geom", "nbinom"), nrow = S, ncol = S, byrow = TRUE)
# creation of an array containing the parameters
param1.matrix = matrix(c(0.6,2,0.4,4,0.7,2,5,0.6,
2,3,0.6,0.6,4,0.3,0.4,4),nrow = S, ncol = S, byrow = TRUE)
param2.matrix = matrix(c(0.8,0,0,2,0,5,0,0.8,
0,0,0.8,0,4,0,0,4),nrow = S, ncol = S, byrow = TRUE)
param.array = array(c(param1.matrix, param2.matrix), c(S,S,2))
# for the reproducibility of the results
set.seed(1)
# simulation of the sequence
seq = simulSM(E = E, NbSeq = 1, lengthSeq = lengthSeq, init = vect.init ,
Ptrans = Pij, distr = distr.matrix, param = param.array,
File.out = "seq.txt")
```

Note that the parameters of the distributions are given in the following way: for example, $f_{13}(\cdot)$ is Geometric distribution with parameter 0.4, while $f_{14}(\cdot)$ is Negative Binomial with parameters 4 and 2. In other words, the parameters of $f_{13}(\cdot)$ are given in the vector param.array $[1, 3,]$ that is equal to $(0.4, 0)$ and the parameters of $f_{14}(\cdot)$ are given in the vector param.array $[1, 4,]$ that is equal to $(4, 2)$, that means that if a distribution has only 1 parameter, the corresponding vector of parameters will have 0 on the second position.

Values : The function **simulSM()** returns a list of simulated sequences. These sequences can be saved in a fasta file by using the parameter File.out.

```
> seq
 [[1]]
    [1] "g" "g" "g" "g" "c" "c" "c" "a" "a" "a" "c" "c" "c" "g"
  [15] "g" "c" "c" "c" "c" "c" "g" "g" "g" "a" "a" "a" "a" "a"
```

Non-parametric simulation: according to distributions given by the user

Now we will consider the simulation according to distributions given by the user.
Parameters : This simulation is carried out by the function **simulSM()**. The different parameters of this function are denoted in the previous subsection (**E, NbSeq, lengthSeq, init, Ptrans, File.out**) 3.2.1, but we will show the difference between the parameters here:
distr : Sojourn time distributions:

- **laws :** Sojourn time distributions introduced by the user:

  - is an array of size $S \times S \times Kmax$.

  where Kmax is the maximum length for the sojourn times.

Example 6. *The R commands below generate three sequences of size 1000, 10000 and 2000 respectively with the finite state space $E = \{a, c, g, t\}$.*

```
## state space
E = c("a","c","g","t")
S = length(E)
```

```
## sequence sizes
lengthSeq3 = c(1000, 10000, 2000)
## creation of the initial distribution
vect.init = c(1/4,1/4,1/4,1/4)
## creation of transition matrix
Pij = matrix(c
    (0,0.2,0.3,0.4,0.2,0,0.5,0.2,0.5,0.3,0,0.4,0.3,0.5,0.2,0),
               ncol=4)
## creation of a matrix corresponding to the conditional
## sojourn time distributions
Kmax = 6
param1.matrix = matrix(c(0.2,0.1,0.3,0.2,0.2,0,0.4,0.2,0.1,
                         0,0.2,0.1,0.5,0.3,0.15,0.05,0,0,
                         0.3,0.2,0.1,0.2,0.2,0),
                       nrow = S, ncol = Kmax, byrow = TRUE)
param2.matrix = matrix(c(0.2,0.1,0.3,0.2,0.2,0,0.4,0.2,0.1,
                         0,0.2,0.1,0.5,0.3,0.15,0.05,0,0,
                         0.3,0.2,0.1,0.2,0.2,0),
                       nrow = S, ncol = Kmax, byrow = TRUE)
param.array = array(c(param1.matrix, param2.matrix), c(S,S,Kmax))
## simulation of 3 sequences without censoring
seqNP3_no = simulSM(E = E, NbSeq = 3, lengthSeq = lengthSeq3, init =
    vect.init, Ptrans = Pij, laws =param.array, File.out = "seqNP3_no.
    txt")
## for the reproducibility of the results
seqNP3_no = read.fasta("seqNP3_no.txt")
seqNP3_no[[1]][1:15]
```

Values : The function **simulSM()** returns a list of simulated sequences. These sequences can be saved in a fasta file by using the parameter File.out.

```
> seqNP3_no[[1]][1:15]
 [1] "c" "c" "g" "g" "g" "g" "g" "g" "c" "c" "c" "a" "a" "a" "a"

```

3.2.2 Estimation of semi-Markov models

In this subsection we explain and illustrate the estimation of a semi-Markov model in the non-parametric cases.

Non-parametric estimation of semi-Markov models

Parameters : The estimation is carried out by the function **estimSM()** and several parameters must be given.

- **file :** Path of the fasta file which contains the sequences from which to estimate.
- **seq :** List of the sequence(s) from which to estimate.
- **E :** Vector of state space of length S.
- **TypeSojournTime :** Type of sojourn time, always equal to "NP" for the non-parametric estimation.

Note that the sequences from which we estimate can be given either as an R list (seq argument) or as a file in fasta format (file argument). The parameter **distr** is always equal to "NP".

Example 7.

```
# data
seqNP3_no = read.fasta("seqNP3_no.txt")
E = c("a","c","g","t")
# estimation of simulated sequences
estSeqNP3= estimSM(seq = seqNP3_no, E = E, distr = "NP", cens.end =
    0, cens.beg = 0)
```

Values : The function **estimSM()** returns a list containing :

- **init :** Vector of size S with estimated initial probabilities of the semi-Markov chain.

```
 > estSeqNP3
 $init
[1] 0.22222222 0.27777778 0.05555556 0.44444444

```

- **Ptrans :** Matrix of size $S \times S$ with estimated transition probabilities of the embedded Markov chain $J = (J_m)_m$.

```
$Ptrans
          [,1]      [,2]      [,3]      [,4]
[1,] 0.0000000 0.1991347 0.4944780 0.3063873
[2,] 0.1973830 0.0000000 0.3046130 0.4980040
[3,] 0.3071290 0.4945768 0.0000000 0.1982942
[4,] 0.3957529 0.2074217 0.3968254 0.0000000


```

- **laws :** Array of size $S \times S \times Kmax$ with estimated values of the sojourn time distributions.

```
estSeqNP3$laws[,,1:2]
 , , 1

          [,1]      [,2]      [,3]      [,4]
[1,] 0.0000000 0.5288736 0.5081741 0.5930881
[2,] 0.6584270 0.0000000 0.6665453 0.4061456
[3,] 0.6679746 0.4676664 0.0000000 0.6624591
[4,] 0.2414634 0.6302999 0.6821622 0.0000000

, , 2

           [,1]       [,2]       [,3]      [,4]
[1,] 0.00000000 0.17324185 0.27630670 0.1553326
[2,] 0.07415730 0.00000000 0.09683291 0.3326653
[3,] 0.05523695 0.26579194 0.00000000 0.1136045
[4,] 0.50081301 0.06153051 0.10162162 0.0000000


```

- **q :** Array of size $S \times S \times Kmax$ with estimated semi-Markov kernel.

```
estSeqNP3$q[,,1]
$q
, , 1

           [,1]      [,2]      [,3]      [,4]
[1,] 0.00000000 0.1053171 0.2512809 0.1817147
[2,] 0.12996230 0.0000000 0.2030384 0.2022621
```

```
[3,] 0.20515435 0.2312969 0.0000000 0.1313618
[4,] 0.09555985 0.1307379 0.2706993 0.0000000

```

3.3 Package HMM

The **HMM** package is a compact package designed for fitting an HMM for a single observation sequence. Here some functions which we can apply for simulating and estimating an HMM:

- **initHMM**: Initialization of HMMs.

  This function initializes a general discrete time and discrete space Hidden Markov Model (HMM). A HMM consists of an alphabet of states and emission symbols. A HMM assumes that the states are hidden from the observer, while only the emissions of the states are observable. The HMM is designed to make inference on the states through the observation of emissions. The stochastic of the HMM is fully described by the initial starting probabilities of the states, the transition probabilities between states and the emission probabilities of the states.

  Usage.

```
initHMM(States, Symbols, startProbs=NULL, transProbs=NULL,
    emissionProbs=NULL)
```

The parameters of the function.

States : Vector with the names of the states.

Symbols : Vector with the names of the symbols.

startProbs : Vector with the starting probabilities of the states. The entries must sum to 1.

transProbs : Stochastic matrix containing the transition probabilities between the states. transProbs is a (number of states)×(number of states).

emissionProbs : Stochastic matrix containing the emission probabilities of the states. emissionProbs is a (number of observation)×(number of observation).

Example 8.

```
# Initialize HMM:
initHMM(c("X","Y"), c("a","b"), c(.3,.7), matrix(c(.9,.1,.1,.9),2),
matrix(c(.3,.7,.7,.3),2))
```

The function return this parameters, we can see the result:

```
$States
[1] "X" "Y"

$Symbols
[1] "a" "b"

$startProbs
  X   Y
0.3 0.7

$transProbs
    to
from   X   Y
   X 0.9 0.1
   Y 0.1 0.9

$emissionProbs
      symbols
states   a   b
     X 0.3 0.7
     Y 0.7 0.3
```

- **simHMM**: Simulate states and observations for a Hidden Markov Model.

  Usage.

```
simHMM(hmm, length)
```

The parameters of the function.

hmm : A Hidden Markov Model.

length : The length of the simulated sequence of observations and states.

Return Value :

The function **simHMM** returns a path of states and associated observations :

states :	The path of states.
observations :	The sequence of observations.

Example 9.

```
# Initialise HMM
hmm = initHMM(c("X","Y"),c("a","b","c"))
# Simulate from the HMM
simHMM(hmm,100)
```

as result we have:

```
> simHMM(hmm, 100)
$states
[1] "X" "X" "Y" "Y" "Y" "Y" "X" "X" "X" "X" "Y" "X" "Y" "Y"
[15] "X" "X" "X" "X" "X" "X" "X" "Y" "Y" "X" "X" "X" "Y" "X"
  .    .
  .    .
[99] "X" "X"

$observation
[1] "b" "b" "c" "c" "c" "c" "a" "b" "a" "b" "b" "b" "c" "c"
[15] "a" "b" "b" "b" "a" "b" "a" "a" "c" "a" "a" "c" "b" "a"
 .     .
 .     .
[99] "a" "b"
```

- **forward** : Computes the forward probabilities.

  The forward probability for state s up to observation at time n is defined as the probability of observing the sequence of observations $r_1, \ldots, r_n$ and that the state at time n is s.

  Usage.

```
forward(hmm,observation)
```

The parameters of the function.

hmm :	A Hidden Markov Model.
observation :	A sequence of observations.

Return Value :

forward : A matrix containing the forward probabilities. The probabilities are given on a logarithmic scale (natural logarithm). The first dimension refers to the state and the second dimension to time.

Example 10.

```
# Initialize HMM
hmm = initHMM(c("A","B"), c("L","R"), transProbs=matrix(c
    (.8,.2,.2,.8),2),
emissionProbs=matrix(c(.6,.4,.4,.6),2))
print(hmm)
# Sequence of observations
observations = c("L","L","R","R")
# Calculate forward probabilities
logForwardProbabilities = forward(hmm, observations)
print(exp(logForwardProbabilities))
```

we see the result:

```
> print(exp(logForwardProbabilities))
        index
states    1     2      3        4
     A  0.3 0.168 0.0608 0.024448
     B  0.2 0.088 0.0624 0.037248

```

- **backward** : Computes the backward probabilities.

  The backward probability for state s and observation at time n is defined as the probability of observing the sequence of observations $r_{n+1}, \ldots, r_T$ under the condition that the state at time n is s.

  Usage.

```
backward(hmm, observation)
```

Return Value :

backward : A matrix containing the backward probabilities. The probabilities are given on a logarithmic scale (natural logarithm). The first dimension refers to the state and the second dimension to time.

Example 11.

```
# Initialize HMM
hmm = initHMM(c("A","B"), c("L","R"), transProbs=matrix(c(.8,.2,.2,.8),2),
emissionProbs=matrix(c(.6,.4,.4,.6),2))
print(hmm)
# Sequence of observations
observations = c("L","L","R","R")
# Calculate backward probabilities
logBackwardProbabilities = backward(hmm, observations)
print(exp(logBackwardProbabilities))
```

we see the result:

```
> print(exp(logBackwardProbabilities))
        index
states        1     2    3 4
     A  0.12416 0.208 0.44 1
     B  0.12224 0.304 0.56 1
```

- **viterbi** : Computes the most probable path of states for a sequence of observations for a given Hidden Markov Model.

Usage.

```
viterbi(hmm, observation)
```

Return Value :

viterbiPath : A vector of strings, containing the most probable path of states.

Example 12.

```
# Initialize HMM
hmm = initHMM(c("A","B"), c("L","R"), transProbs=matrix(c
    (.6,.4,.4,.6),2),emissionProbs=matrix(c(.6,.4,.4,.6),2))
print(hmm)
# Sequence of observations
observations = c("L","L","R","R")
# Calculate Viterbi path
viterbi = viterbi(hmm,observations)
print(viterbi)
```

we see the result:

```
> print(viterbi)
[1] "A" "A" "B" "B"
```

- **baumWelch** : Inferring the parameters of a Hidden Markov Model via the Baum-Welch algorithm.

  For an initial Hidden Markov Model (HMM) and a given sequence of observations, the Baum-Welch algorithm infers optimal parameters to the HMM. Since the Baum-Welch algorithm is a variant of the Expectation-Maximization algorithm (EM), the algorithm converges to a local solution which might not be the global optimum.

  Usage.

```
baumWelch(hmm, observation , maxIterations=100,delta=1E-9,
    pseudoCount=0)
```

The parameters of the function.

hmm : A Hidden Markov Model.

observation : A sequence of observations.

maxIterations : The maximum number of iterations in the Baum-Welch algorithm.

pseudoCount : Adding this amount of pseudo counts in the estimation-step of the Baum-Welch algorithm.

Return Values :

hmm : The inferred HMM. The representation is equivalent to the representation in **initHMM**.

Example 13.

```
# Initialize HMM\|
hmm = initHMM(c("A","B"),c("L","R"),
transProbs=matrix(c(.9,.1,.1,.9),2), emissionProbs=matrix(c
    (.5,.51,.5,.49),2))
print(hmm)
# Sequence of observation
a = sample(c(rep("L",100),rep("R",300)))
b = sample(c(rep("L",300),rep("R",100)))
observation = c(a,b)
# Baum-Welch
bw = baumWelch(hmm,observation,10)
print(bw$hmm)
> observation
[1] "R" "R" "R" "L" "L" "R" "L" "R" "R" "L" "R" "R" "L" "L"
[15] "R" "R" "R" "R" "R" "R" "R" "R" "L" "R" "R" "R" "R" "R"
   .        .          .                .                  .
   .        .          .                .                  .
[785] "L" "L" "R" "L" "L" "L" "L" "L" "L" "L" "L" "L" "L" "R"
[799] "L" "L"
> print(bw$hmm)
$States
[1] "A" "B"

$Symbols
[1] "L" "R"

$startProbs
  A   B
0.5 0.5

```

```
$transProbs
       to
from              A             B
    A 9.974891e-01 0.002510914
    B 5.337632e-06 0.999994662

$emissionProbs
          symbols
states            L          R
        A 0.2500659 0.7499341
        B 0.7486748 0.2513252
```

3.4 Package hsmm

The package hsmm provides tools for performing HSMM analysis, which are commonly required when working with this model. The main requirements are:

- the simulation of sequences of states and observations.
- the estimation of model parameters.
- the analysis of the underlying state sequence.

3.4.1 Simulation of observation and state sequences

To obtain a first understanding of the nature and properties of HSMMs, the simulation of sequences of states and observations is a helpful tool. For given model specifications, this is carried out by the function **hsmm.sim()**.

Usage.

```
hsmm.sim(n, od, rd, pi.par, tpm.par, od.par, rd.par, M = NA, seed = NULL
    )
```

The parameters of the function.

- **n :** Positive integer containing the number of observations to simulate.
- **od :** Character containing the name of the conditional distribution of the observations.

- **rd :** Character containing the name of the runlength distribution (or sojourn time distribution).
- **pi.par :** Vector of length J containing the values for the initial probabilities of the semi-Markov chain.
- **tpm.par :** Matrix of dimension $J \times J$ containing the parameter values for the transition probability matrix of the embedded Markov chain. The diagonal entries must all be zero, absorbing states are not permitted.
- **rd.par :** List with the values for the parameters of the runlength distributions.
- **od.par :** List with the values for the parameters of the conditional observation distributions.
- **M :** Positive integer containing the maximum runlength.
- **seed :** Seed for the random number generator (integer).

Example 14. *The R commands given below generate a sequence of length n = 2000 from a HSMM with poisson observation distributions, Poisson (discrete distribution) runlength distributions, and three hidden states.*

```
## Setting up the parameter values:
# Initial probabilities of the semi-Markov chain:
pipar <- rep(1/3, 3)
# Transition probabilities:
# (Note: For two states, the matrix degenerates, taking 0 for the
# diagonal and 1 for the off-diagonal elements.)
tpmpar <- matrix(c(0, 0.5, 0.5, 0.7, 0, 0.3, 0.8, 0.2, 0), 3, byrow =
    TRUE)
# sojourn time distribution:
rdpar <- list(lambda = c(0.98, 0.99, 1))
# Observation distribution:
odpar <- list(lambda = c(0.5, 0.6, 0.8))
# Invoking the simulation:
sim <- hsmm.sim(n = 2000, od = "pois", rd = "pois",
pi.par = pipar, tpm.par = tpmpar,
rd.par = rdpar, od.par = odpar, seed = 3539)
# The first 15 simulated observations:
round(sim$obs[1:15], 3)
# The first 15 simulated states:
sim$path[1:15]
```

Return Value: The function **hsmm.sim()** returns a list containing the simulated sequence of observations and the simulated state sequence. The simulated observations and states are accessed with

```
> round(sim$obs[1:15], 3)
 [1] 0 0 0 0 0 0 0 2 2 1 0 2 1 1 0
> sim$path[1:15]
 [1] 2 3 3 3 3 2 2 3 1 2 2 1 1 2 1
```

3.4.2 Maximum likelihood estimation of the model parameters

The estimation is carried out by the function **hsmm()**. The arguments for this function are similar to **hsmm.sim()** with the only difference being in the parameter specification (i.e. the arguments ending on **.par**). In the case of **hsmm()**, the arguments **pi.par, tpm.par, rd.par, od.par** specify the starting values for the parameter estimation. The function hsmm fits a hidden semi-Markov model using the EM algorithm for parameter estimation. The estimation algorithms are based on the right-censored approach initially described in Guedon (2003).

Usage.

```
hsmm(x, od, od.par, rd , rd.par, pi.par, tpm.par, M = NA, Q.max,
    epsilon, censoring, prt, detailed, r.lim, p.log.lim , nu.lim)
```

The parameters of the function.

- **x :** The observations as a vector of length T.
- **od :** Character with the name of the conditional distribution of the observations. The following distributions are currently implemented :

"bern" = Bernoulli
"norm" = Normal
"pois" = Poisson
"t" = Student's t

- **rd :** Character with the name of the runlength distribution (or sojourn time, dwell time distribution). The following distributions are currently implemented :

  "nonp" = Non-parametric
  "geom" = Geometric
  "nbinom" = Negative Binomial
  "log" = Logarithmic
  "pois" = Poisson

- **pi.par :** Vector of length J with the initial values for the initial probabilities of the semi-Markov chain.
- **tpm.par :** Matrix of dimension $J \times J$ with the initial values for the transition probability matrix of the embedded Markov chain. The diagonal entries must all be zero, absorbing states are not permitted.
- **rd.par :** List with the initial values for the parameters of the runlength distributions.
- **od.par :** List with the initial values for the parameters of the conditional observation distributions.
- **M :** Positive integer containing the maximum runlength.
- **Q.max :** Positive integer containing the maximum number of iterations.
- **epsilon :** Positive scalar giving the tolerance at which the relative change of log-likelihood is considered close enough to zero to terminate the algorithm.

Example 15. *The following example illustrates the usage of **hsmm()** for parameter estimation.*

```
# Simulating observations:
pipar <- rep(1/3, 3)
tpmpar <- matrix(c(0, 0.5, 0.5, 0.7, 0, 0.3, 0.8, 0.2, 0), 3, byrow =
    TRUE)
rdpar <- list(lambda = c(0.98, 0.99, 1))
odpar <- list(lambda = c(0.5, 0.6, 0.8))
sim <- hsmm.sim(n = 2000, od = "pois", rd = "pios",
pi.par = pipar, tpm.par = tpmpar,
rd.par = rdpar, od.par = odpar, seed = 3539)
# Executing the EM algorithm :
```

```
fit <- hsmm(sim$obs, od = "pois", rd = "pois",
pi.par = pipar, tpm.par = tpmpar,
od.par = odpar, rd.par = rdpar)
# The log-likelihood:
fit$logl
# the estimated parameters:
fit$para
# For comparison, the estimated parameters separately together with the true parameter values are given below.
# Transition probability matrix:
tpmpar
fit$para$tpm
# Observation distribution:
odpar
fit$para$od
# Runlength distribution:
rdpar
fit$para$rd
```

The function **hsmm()** returns a list containing the output. For example, the observed data log likelihood is returned in the logl entry :

```
> # The log-likelihood:
> fit$logl
[1] -2085.732
```

The estimated parameters are given in the para entry, where each component (TPM, runlength distribution and observation distribution) is given in a sub-list.

```
> fit$para
$pi
[1] 9.999483e-01 5.173197e-05 2.331468e-15

$tpm
          [,1]      [,2]      [,3]
[1,] 0.0000000 0.5015614 0.4984386
[2,] 0.7103073 0.0000000 0.2896927
[3,] 0.8012183 0.1987817 0.0000000

$rd
$rd$lambda
[1] 0.9781819 0.9963024 0.9992340
```

```


$od
$od$lambda
[1] 0.4889226 0.5976942 0.8298671
```

For comparison, the estimated parameters separately together with the true parameter values:

```
> # Transition probability matrix:
> tpmpar
     [,1] [,2] [,3]
[1,]  0.0  0.5  0.5
[2,]  0.7  0.0  0.3
[3,]  0.8  0.2  0.0
> fit$para$tpm
          [,1]      [,2]      [,3]
[1,] 0.0000000 0.5015614 0.4984386
[2,] 0.7103073 0.0000000 0.2896927
[3,] 0.8012183 0.1987817 0.0000000
> # Observation distribution:
> odpar
$lambda
[1] 0.5 0.6 0.8

> fit$para$od
$lambda
[1] 0.4889226 0.5976942 0.8298671

> # Runlength distribution:
> rdpar
$lambda
[1] 0.98 0.99 1.00

> fit$para$rd
$lambda
[1] 0.9781819 0.9963024 0.9992340
```

3.4.3 Inference on the underlying state sequence

Inference on the hidden states for given observations and model specifications of a hidden semi-Markov model. The function **hsmm.viterbi()** carries out the Viterbi algorithm. It derives the most probable state sequence by a dynamic programming technique. This procedure is often termed "global decoding". The arguments for this function are similar to **hsmm.sim()** and **hsmm**.

Usage.

```
hsmm.viterbi(x, od, rd, pi.par, tpm.par, od.par, rd.par, M = NA)
```

Example 16.

```
# Simulating observations:
pipar <- rep(1/3, 3)
tpmpar <- matrix(c(0, 0.5, 0.5, 0.7, 0, 0.3, 0.8, 0.2, 0), 3, byrow =
    TRUE)
rdpar <- list(lambda = c(0.98, 0.99, 1))
odpar <- list(lambda = c(0.5, 0.6, 0.8))
sim <- hsmm.sim(n = 2000, od = "pois", rd = "pois",
pi.par = pipar, tpm.par = tpmpar,
rd.par = rdpar, od.par = odpar, seed = 3539)
# Executing the Viterbi algorithm:
fit.vi <- hsmm.viterbi(sim$obs, od = "pois", rd = "pois",
pi.par = pipar, tpm.par = tpmpar,
od.par = odpar, rd.par = rdpar)
# The first 15 values of the resulting path :
fit.vi$path[1:15]
# For comparison the real/simulated path (first 15 values):
sim$path[1:15]
```

The function return a vector of length T containing the most probable path of the underlying states:

```
> fit.vi$path[1:15]
  [1] 1 1 1 2 2 1 1 3 3 1 1 3 3 3 1
> # For comparison the real/simulated path (first 15 values):
> sim$path[1:15]
  [1] 2 3 3 3 3 2 2 3 1 2 2 1 1 2 1
```

Conclusion

In this work we have considered the popular hidden Markov model (HMM), and as an extension a hidden semi-Markov model (HSMM) which allows the underlying stochastic process to be a semi-Markov chain. Each state has variable duration and a number of observations being produced while in the state. This makes it suitable for use in a wider range of applications. Its forward-backward, viterbi and Baum-welch algorithms can be used to estimate/update the model parameters, determine smoothed probabilities, evaluate goodness of an observation sequence fitting to the model, and find the best state sequence of the underlying stochastic process. For HSMM model, the nonparametric maximum likelihood estimators for the characteristics of such a model have nice asymptotic properties, namely consistency and asymptotic normality. An R implementation of this models has been presented.

Since the HSMM was initially introduced in 1980 for machine recognition of speech, it has been applied in thirty scientific and engineering areas, such as speech recognition/synthesis, human activity recognition/prediction, handwriting recognition, functional MRI brain mapping, and network anomaly detection. There are about three hundred papers published in the literature.

Bibliography

[1] N.B. Amara, A. Belaid, Printed PAW recognition based on planar hidden Markov models, in: Proceedings of the 13th International Conference on Pattern Recognition, vol. 2, 1996, pp. 220-224.

[2] V. Barbu and N. Limnios. Discrete time semi-Markov processes for reliability and survival analysis, Communications in statistic theory and methods. Taylor and Francis publisher 2004.

[3] V. Barbu and N. Limnios. Empirical estimation for discrete time semi-Markov processes with applications in reliability. J.Nonparametr.Statist. 2006a.

[4] V. Barbu and N. Limnios. Semi-Markov chains and Hidden Semi Markov Models toward Applications vol.191, Spring, New York.(2008a).

[5] L.E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. Ann. Math. Statist., 37:1554-1563, 1966.

[6] L. E. Baum, G. R. Sell : Growth functions for transformations on manifolds, Pac.J. Math., vol. 27, no.2, 211-227 ,1968.

[7] P.J. Bickel and Y. Ritov. Inference in hidden Markov models i: local asymptotic normality in the stationary case. Bernoulli, 2(3):199-228, 1996.

[8] P.J. Bickel, Y. Ritov, and T. Ryd´en. Asymptotic normality of the maximum likelihood estimator for general hidden Markov models. Ann. Statist., 26:1614-1635, 1998.

[9] P. Billingsley. The Lindeberg-Levy theorem for martingales. Proc.Amer. Math. Soc. 1961b.

[10] P. Billingsley. Convergence of Probability Measures. Wiley, New York, 2nd edition, 1999.

[11] P. Billingsley. Probability and Measure. Wiley, New York, 3rd edition, 1995.

[12] G. Churchill. Hidden Markov chains and the analysis of genome structure. Comput.Chem., 16:107-115, 1992.

[13] R. Douc. Non singularity of the asymptotic Fisher information matrix in hidden Markov models. *École* Polytechnique, preprint, 2005.

[14] R. Durbin, S.R. Eddy, A. Krogh, and G.J. Mitchison. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press,Cambridge, 1998.

[15] J.D. Ferguson. Variable duration models for speech. In Proc. of the Symposium on the Application of Hidden Markov Models to Text and Speech, pages 143-179.Princeton, NJ, 1980.

[16] A. Gut. Stopped Random Walks. Limit Theorems and Applications,volume 5 of Applied Probability. A Series of the Applied Probability Trust. Springer, New York, 1988.

[17] V. Krishnamurthy, J.B. Moore, S.H. Chung, Hidden fractal model signal processing, Signal Processing 24 (2) (Aug. 1991) 177-192.

[18] D. Kulp, D. Haussler, M.G. Reese, F.H. Eeckman, A generalized hidden Markov model for the recognition of human genes in DNA, in: Proc. 4th Int.Conf. Intell. Syst. Molecular Bio., 1996, pp. 134-142.

[19] A. Kundu, Y. He, M. Y. Chen, Efficient utilization of variable duration information in HMM based HWR systems, in: Proceedings of International Conference on Image Processing, 1997, vol. 3, 26-29 Oct. 1997, pp. 304-307.

[20] B.G. Leroux. Maximum-likelihood estimation for hidden Markov models. Stochastic Process Appl., 40:127-143, 1992.

[21] P. Levy. Processus semi-Markoviens. In Proc. of International Congress of Mathematics, Amsterdam, 1954.

[22] N. Limnios and Oprisan, G, 2001, Semi-Markov Processes and Reliability (Boston: Birkhäuser).

[23] E. Marcheret, M. Savic, Random walk theory applied to language identification, in: Proc. of 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-97, vol.2, 21-24 April 1997, pp.1119-1122.

[24] E.H. Moore, R. Pyke, Estimation of the transition distributions of a Markov renewal process, Ann. Inst. Stat. Math. 20 (1968).

[25] T. Petrie. Probabilistic functions of finite state Markov chains. Ann. Math. Statist.,40:97-115, 1969.

[26] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE, 77:257-286, 1989.

[27] W.L. Smith. Regenerative stochastic processes. Proc. R. Soc. Lond. Ser. A Math. Phys. Eng., 232:6-31, 1955.

[28] S. Trevezas and N. Limnios. Maximum likelihood estimation for general hidden semi-Markov processes with backward recurrence time dependence. to appear in J. of Mathematical Sciences, 2008b.

[29] S.V. Vaseghi, Hidden Markov models with duration-dependent state transition probabilities, Electronics Letters 27 (8) (April 1991) 625-626.

[30] S-Z Yu, H. Kobayashi, An efficient forward-backward algorithm for an explicit duration hidden Markov model, IEEE Signal Processing Letters 10 (1)(Jan. 2003) 11-14.