MOX-Report No. 21/2019

Hidden Markov Models for multivariate functional data

Martino, A.; Guatteri, G.; Paganoni, A.M.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it http://mox.polimi.it

Hidden Markov Models for multivariate functional data

Andrea Martino[1], Giuseppina Guatteri[1] and Anna Maria Paganoni[1]

[1]Department of Mathematics, Politecnico di Milano, Milan, Italy

Abstract

Hidden Markov Models (HMMs) are a very popular tool used in many fields to model time series data. In this paper we want to extend the usual HMM framework, where the observed objects are univariate or multivariate data, to the case of functional data. In particular, since we have a sequence of multivariate curves that evolves in time, we want to model the temporal structure of the system using HMMs. The functional observations, which rely on the statistical tools related to Functional Data Analysis (FDA), are linked to the state of the HMM according to a similarity function, which depends on some metric in Hilbert spaces. We first assess our results in a simulation setting and then we apply our model to a case study regarding the climate.
Keywords: Functional Data; Hidden Markov Models.

1 Introduction

Hidden Markov Models (HMMs) represent a well-known method for the study of time series involving sequences of data, widely used in many fields like biostatistics ([9]), bioinformatics ([6]) and finance ([11]). They are a generalization of mixture models, where the hidden variables controlling the mixture components are related through a Markov process, see [13] for further details. In the literature of HMMs, there are several examples where the outcome consists of univariate or multivariate data, with both discrete and continuous observations; in particular, in [4] a very general definition of such processes is provided. In this paper, we want to extend the usual HMM algorithms from the finite dimensional framework to the infinite dimensional one. Therefore, we focus on the functional setting, where each observed data is considered as a multivariate random curve, that can be also seen as the realization of a stochastic process taking values in $\mathbb{R}^n, n \geq 1$.

The natural context to develop the statistical models and tools to describe this kind of data is the *Functional Data Analysis* (FDA) (see, e.g. [14], [15], [7], [8]). Working with functional data can be a difficult task because of the dimensionality of the spaces of the data; moreover, the usual HMM requires the definition of a probability density that generates

the observations, which may be lacking for functional random processes. Therefore, since we want to consider the most general case without making any assumptions on the law of the process that generated the data, we construct a similarity function built on distances between curves to evaluate the emission of an observation by a certain state.

We consider a hidden Markov chain evolving in time where each state emits a multivariate random curve and we solve two problems. First, we estimate the parameters of the underlying Markov process to understand the time series system that generated the data; then we solve a clustering problem by finding the best sequence of states that generated the data in order to classify the curves in clusters.

The paper is organized as follows: in Section 2 we present the model, adding some information about the theory of HMMs and functional data. In Section 3 we present a simulation study to assess the performance of the model while in Section 4 we see a case study application to a dataset regarding a climate problem. Finally, in Section 5 we give some discussion and conclusions. All the analysis have been carried out using the statistical software `R` ([12]) and the codes are available upon request.

2 The model

The aim of this paper is to consider and study a proper Hidden Markov Model (HMM) in the multivariate functional framework. Let us consider a multivariate random curve $\mathbf{X} = \{\mathbf{X}(t)\}_{t\in I} = \{X_1(t), \dots, X_J(t)\}_{t\in I}$, with $J \geq 1$ and I compact interval of $\mathbb{R}$. The scalar product between two elements $\mathbf{a}, \mathbf{b} \in L^2(\Omega \times I; \mathbb{R}^J)$ is defined as follows:

$$\langle \mathbf{a}, \mathbf{b} \rangle = \mathbb{E}\left[\sum_{j=1}^{J} \int_I a_j(t) b_j(t) dt\right].$$

Let $\boldsymbol{\mu}(t) = \mathbb{E}[\mathbf{X}(t)]$ be the mean function of $\mathbf{X}$ for almost all $t \in I$ and $v(s,t) = \mathbb{C}\mathrm{ov}\,[\mathbf{X}(s), \mathbf{X}(t)]$ be its covariance kernel, i.e. $v(s,t)$ is a $J \times J$ matrix of functions such that $v_{j_1 j_2}(s,t) := \mathbb{C}\mathrm{ov}\,[X_{j_1}(s), X_{j_2}(t)]$ for any $j_1, j_2 \in \{1, \dots, J\}$ and $s, t \in I$.

We define a Hidden Markov Model, see [4], as a bivariate process $\{(Q_k,$ $\{\mathbf{X}_k(t)\}_{t\in I})\}_{k\geq 0}$ on a given probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where $\{\mathbf{X}_k(t)\}_{t\in I}$ is a multivariate random curve and $\{Q_k\}_{k\geq 0}$ is a Markov chain with a discrete and finite state space $\{s_1, \dots, s_N\}$, with $N \geq 1$, transition matrix $A = \{a_{ij}\} = \mathbb{P}(Q_k = s_j | Q_{k-1} = s_i)$ and initial distribution $\boldsymbol{\nu}$, where $\nu_i = \mathbb{P}(Q_0 = s_i)$. Given the process $\{Q_k\}_{k\geq 0}$, $\{\{\mathbf{X}_k(t)\}_{t\in I}\}_{k\geq 0}$ is a sequence of conditionally independent multivariate functions and $\{\mathbf{X}_k(t)\}_{t\in I}$ only depends on Q_k for each k . We denote the emission function of $\mathbf{X}_k$ conditionally on the event $\{Q_k = s_i\}$ with $b_i(\,\cdot\,; \boldsymbol{\mu}_i)$, for any $i = 1, \dots, N$, where $\boldsymbol{\mu}_i$ is a functional parameter representing the mean of the curves emitted by state s_i. We can completely define our HMM with the set of parameters $\lambda = (\boldsymbol{\nu}, A, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_N)$.

In this work, we use distances between functions to construct the emission functions $b_i(\,\cdot\,; \boldsymbol{\mu}_i)$, $i = 1, \dots, N$. We will assume that, for each state s_i, the emission function can be written as

$$b_i(\,\cdot\,; \boldsymbol{\mu}_i) = h\big(d(\,\cdot\,, \boldsymbol{\mu}_i)\big), \qquad i = 1, \dots, N \tag{2.1}$$

where $h : \mathbb{R} \to \mathbb{R}$ is a function that transforms the distance into a similarity measure. In particular, in this work we will use the function $h(t) = 1/t^2$ and the L^2 distance that, in the multivariate functional framework, is defined as follows:

$$d_{L^2}(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\| = \sqrt{\sum_{l=1}^{J} \int_I (a_l(t) - b_l(t))^2 dt}.$$

An important step of our algorithm is the initialization. Since we want our algorithm to be as robust as possible, we perform a functional k-means algorithm on the dataset of curves to find the initial centroids, see [17] for further details. The initialization step consists of an iterative procedure alternating a step where each multivariate curve is assigned to a cluster and a step where the centroids of the clusters are computed. Specifically, after a random selection of a set of N fixed initial centroids $\{\boldsymbol{\mu}_1^{(0)}(t), \ldots, \boldsymbol{\mu}_N^{(0)}(t)\}$ the algorithm iteratively repeats the two steps described before. Formally, at the m-th iteration of the initialization step, $m \geq 1$, in the algorithm:

1. each curve is assigned to the cluster whose centroid minimizes the L^2 distance. The m-th cluster assignment $C_i^{(m)}$ for the i-th statistical unit, $i = 1, \ldots, n$ is

$$C_i^{(m)} := \underset{l=1,\ldots,N}{\operatorname{argmin}} \, d_{L^2}(\mathbf{X}_i, \boldsymbol{\mu}_l^{(m-1)});$$

2. the centroids for the clusters are computed as

$$\boldsymbol{\mu}_l^{(m)} := \underset{\boldsymbol{\mu} \in L^2(\Omega \times I; \mathbb{R}^J)}{\operatorname{argmin}} \sum_{i:C_i^{(m)}=l} d_{L^2}(\mathbf{X}_i, \boldsymbol{\mu})^2, \qquad l = 1, \ldots, N.$$

   In our case, we can rewrite the equation for any state $l = 1, \ldots, N$ as:

$$\boldsymbol{\mu}_{jl}^{(m)} := \frac{1}{n_l} \sum_{i:C_i^{(m)}=l} \mathbf{X}_i, \qquad j = 1, \ldots, J.$$

   where n_l is the number of curves assigned to the l-th cluster in the step m.

After obtaining the same cluster assignments at two subsequent iterations, the initialization step ends and we obtain the preliminary estimates of the functional parameters $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_N\}$ for the states of the HMM.

Let us denote by $\mathbf{x}$ an output sequence of observation functions of the HMM and with $\mathcal{L}(\lambda|\mathbf{x})$ the objective function of all the parameters of the model given $\mathbf{x}$. In the literature of HMMs, there are usually three problems to tackle (see, e.g., [20] and [13]):

1. find $\mathcal{L}(\lambda|\mathbf{x})$ for the realization $\mathbf{x}$;
2. find the set of parameters $\lambda^* = \underset{\lambda}{\operatorname{argmax}} \, \mathcal{L}(\lambda|\mathbf{x})$;

3. find the best state sequence $Q = (Q_1, \ldots, Q_K)$ that explains $\mathbf{x}$, given $\mathbf{x}$ and λ.

As usually done in the literature, to address these problems we use the forward-backward procedure, the Baum-Welch algorithm and the Viterbi algorithm, respectively. We solve for the forward variables $\alpha_k(j)$ inductively, as follows:

(1) initialization: $\alpha_1(i) = \nu_i b_i(\mathbf{x}_1; \boldsymbol{\mu}_i)$, for $1 \leq i \leq N$;

(2) induction: $\alpha_{k+1}(i) = \left[\sum_{j=1}^{N} \alpha_k(j) a_{ji}\right] b_i(\mathbf{x}_{k+1}; \boldsymbol{\mu}_i)$, for any $1 \leq k \leq K-1$ and $1 \leq i \leq N$;

(3) termination: $\mathcal{L}(\lambda|\mathbf{x}) = \sum_{j=1}^{N} \alpha_K(j)$,

where the emission functions $b_i(\cdot; \boldsymbol{\mu}_i)$ are defined in 2.1. In a similar way, we compute the backward variables $\beta_k(i)$:

(1) initialization: $\beta_K(i) = 1$, for $1 \leq i \leq N$;

(2) induction: $\beta_k(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{x}_{k+1}; \boldsymbol{\mu}_j) \beta_{k+1}(j)$, for any $1 \leq k \leq K-1$, $1 \leq i \leq N$;

(3) termination: $\mathcal{L}(\lambda|\mathbf{x}) = \sum_{i=1}^{N} \beta_1(i) \nu_i b_i(\mathbf{x}_1; \boldsymbol{\mu}_i)$.

To avoid computational problems we apply a procedure in both the forward and backward step, well known as "scaling" in the HMM literature, to successfully implement the estimation of all the parameters of the HMM (see for instance [13] for further details). Separately, both procedure compute $\mathcal{L}(\lambda|\mathbf{x})$, but we need them together to find the model λ that maximizes the function.

The sequence of states of a HMM is not observed, so the usual approach, see [5], consists in treating the states as missing data and apply an EM algorithm to find the best estimates of the parameters. This algorithm is known in the literature as Baum-Welch algorithm (see for instance [1], [2], [19], [3]).

To fully describe our algorithms, we need to introduce two further quantities. We define $\xi_k(i,j)$, the probability of being in state s_i at time k, and state s_j at time $k+1$, given the model and the observations, i.e.

$$\xi_k(i,j) \;=\; \mathbb{P}\left(Q_k = s_i, Q_{k+1} = s_j \mid \mathbf{X}_1 = \mathbf{x}_1, \ldots, \mathbf{X}_k = \mathbf{x}_k, \lambda\right)$$

and the probability of being in the state s_i at time k, given the observations and the model as

$$\gamma_k(i) \;=\; \mathbb{P}\left(Q_k = s_i \mid \mathbf{X}_1 = \mathbf{x}_1, \ldots, \mathbf{X}_k = \mathbf{x}_k, \lambda\right) \;=\; \sum_{j=1}^{N} \xi_k(i,j),$$

As in [20], we can write the logarithm of the objective function of our model as

$$\log(\mathcal{L}(\lambda|\mathbf{x})) = \underbrace{\sum_{j=1}^{N} \gamma_1(j) \log \nu_j}_{\text{term 1}} + \underbrace{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\sum_{k=2}^{K} \xi_k(i,j)\right) \log a_{ij}}_{\text{term 2}} + \underbrace{\sum_{j=1}^{N}\sum_{k=1}^{K} \gamma_k(j) \log b_i(\mathbf{x}_k; \boldsymbol{\mu}_j)}_{\text{term 3}}. \tag{2.2}$$

Using this expression, we apply an EM algorithm to compute all the parameters of the HMM by iteratively repeating the following two steps:

- **E step** replace the quantities $\xi_k(i,j)$ and $\gamma_j(k)$ with their conditional expectations given the current parameter estimates and the observations;
- **M step** maximize the function in Eq. (2.2). Each term of the expression depends on different parameters, so it can be split in three parts. We focus our work on the maximization of the third term, since for the maximization of the first two terms we follow the approach described in [20].

Even though until now our formulas only consider a single observation sequence, they can be extended to the more general case of multiple observations. Let us denote the set of L observation sequences as

$$\mathcal{X} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(L)})$$

where $\mathbf{x}^{(l)} = (\mathbf{x}_1^{(l)}\, \mathbf{x}_2^{(l)} \cdots \mathbf{x}_K^{(l)})$ is the l-th observation sequence of length K_l. We assume all the sequences to be independent from each other; our goal is to adjust the parameters of the model λ to maximize the following function:

$$\mathcal{L}(\lambda|\mathcal{X}) = \prod_{l=1}^{L} \mathcal{L}(\lambda|\mathbf{x}^{(l)}).$$

The term we want to maximize, in the setting of multiple sequences, becomes

$$\text{term 3} = \sum_{l=1}^{L}\sum_{k=1}^{K_l}\sum_{j=1}^{N} \gamma_k(j) \log b_i(\mathbf{x}_k; \boldsymbol{\mu}_j). \tag{2.3}$$

Since we want to estimate the functional parameters $(\mu_{ij})_{i=1,\dots,N;j=1,\dots,J}$ of the states of the HMM, we compute for every state i and every component j

$$\widehat{\mu}_{ij} = \underset{\mu_{ij}}{\operatorname{argmax}} \sum_{l=1}^{L}\sum_{k=1}^{K_l} \gamma_k(j) \log b_i(x_{lk}; \mu_{ij}).$$

To perform this step, we extend all the estimators commonly used in the functional data framework into the theory of functional HMM. Formally, let us denote with $\boldsymbol{\chi}$ the mean

function of the sequence $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_K$ and consider an estimator $\widehat{\boldsymbol{\chi}}$ of $\boldsymbol{\chi}$. For instance, if $\boldsymbol{\chi}$ is the multivariate functional mean of $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_K$, we estimate it as $\widehat{\boldsymbol{\chi}} = \frac{1}{K}\sum_{i=1}^{K} \boldsymbol{X}_i$. Therefore, for every state $i = 1, \ldots, N$, the HMM extension can be written as

$$\widehat{\boldsymbol{\chi}} = \frac{\sum_{l=1}^{L}\sum_{k=1}^{K_l} \gamma_k(i)\, \boldsymbol{X}_k}{\sum_{l=1}^{L}\sum_{k=1}^{K_l} \gamma_k(i)}.$$

Finally, the third problem regarding HMM, as we stated before, consists in finding the best state sequence $Q = (Q_1, \ldots, Q_K)$ that explains a certain observation. To solve this problem we use the Viterbi algorithm , see [18]. Let us define the quantity

$$\delta_k(i) = \max_{Q_1,\ldots,Q_{k-1}} \mathbb{P}(Q_1, Q_2, \ldots, Q_k = s_i, \mathbf{x}_1, \ldots, \mathbf{x}_k|\lambda)$$

which is the highest probability on a single path at time k by taking into account the partial sequence $\mathbf{x}_1, \ldots, \mathbf{x}_k$. We solve by induction as follows

$$\delta_{k+1}(j) = \max_i[\delta_k(i)a_{ij}] \cdot b_j(\mathbf{x}_{k+1})$$

At this point, we perform a procedure similar to the forward one and we can retrieve the best state sequence by keeping track of the argument that maximizes this last expression (see [13] for further details).

3 Simulation Studies

We generate three samples of length $n = 2000$ of realizations on a grid of 100 points for three independent bivariate random curves $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ in $L^2(\Omega \times I; \mathbb{R}^J)$, with $J = 2$. Each sample is emitted from a different state of a 3-state HMM having the following parameters:

- **State 1**: $\nu_1 = 1, a_{11} = 0.6, a_{12} = 0.3, a_{13} = 0.1$, $\boldsymbol{\mu}_1(t) = \begin{pmatrix} t(1-t) \\ 2t \end{pmatrix}$;
- **State 2**: $\nu_2 = 0, a_{21} = 0.1, a_{22} = 0.8, a_{23} = 0.1$, $\boldsymbol{\mu}_2(t) = \begin{pmatrix} t^2(1-t) \\ t^2 \end{pmatrix}$;
- **State 3**: $\nu_3 = 0, a_{31} = 0, a_{32} = 0, a_{33} = 1$, $\boldsymbol{\mu}_3(t) = \begin{pmatrix} t(1-t)^2 \\ \frac{1}{2}t^3 \end{pmatrix}$.

where $\boldsymbol{\nu} = (\nu_i)$ is the vector of the initial probabilities of the state, $A = (a_{ij})$ is the transition matrix and $\boldsymbol{\mu}_i(t), i = 1, \ldots, N$, represent the real means of each sample. For each state, the sample is generated using the same exponential covariance kernel $C(s,t) = ae^{-b|s-t|}$, $a = 0.1$, $b = 0.3$, using the `R` package `roahd` [16]. In Fig. 1 we show a plot of the simulated curves (first component on the left, second component on the right) with a different colour for each group. The first problem consists in the choice of the number of states, since it is a priori unknown. We begin by running our algorithm for $N = 2, \ldots, 5$ number of states and

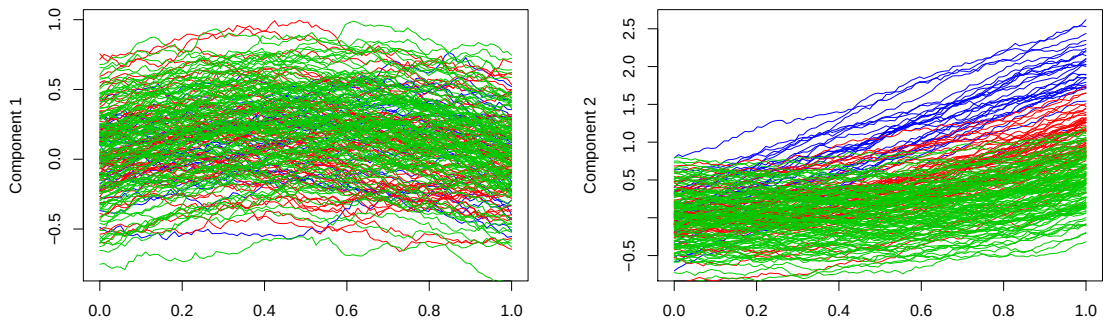

Figure 1: Plots of a subset of 200 bivariate curves for the simulated data.

by computing each time the AIC and BIC criteria to choose the best model in terms of the number of states. In particular, we compute the values as follows:

$$\text{AIC} = -2\log(\mathcal{L}(\lambda|\mathbf{x})) + 2p \qquad \text{BIC} = -2\log(\mathcal{L}(\lambda|\mathbf{x})) + p\log(n) \tag{3.1}$$

where p is the number of parameters of the HMM. As we see in Fig. 2, both criteria reach the minimum value for $N = 3$ states; from now on we choose this value as the "optimal" number of states for the HMM. After choosing the number of states, we summarize our results along 100 repetitions of our algorithm to estimate the parameters of the HMM. To have a better understanding of the results, we compute the mean square error (MSE) and the standard deviation (SD) of the estimates and we show the obtained results in Tab. 1. As we can see, all the parameters are very well estimated, both in terms of mean and standard deviation of the parameters.

Moreover, we can obtain some further information about the clustering structure of our data. Specifically, we use our model and apply the Viterbi algorithm on the output obtained from the Baum-Welch algorithm, to estimate the best state sequence and compare it with the output of the k-means algorithm, based on the same distance. In particular, in Tab. 2 we compare the Correct Classification Rate (CCR) of the number of curves obtained by applying both methods along with the MSEs of the estimates of functional parameters of a state $\widehat{\boldsymbol{\mu}}_i$, $i = 1, 2, 3$, i.e. the values of the distances between the real and the estimated means, computed over 100 replications of the algorithm. By comparing the results, we see obvious advantages of our method, since the CCR is higher and all the MSEs and standard deviations are smaller. We conclude that, not only our method is able to detect the temporal structure behind the sequences of functional data and estimate all the parameters of the underlying hidden states but, applying the Viterbi algorithm, we can also cluster the curves with good values of accuracy.

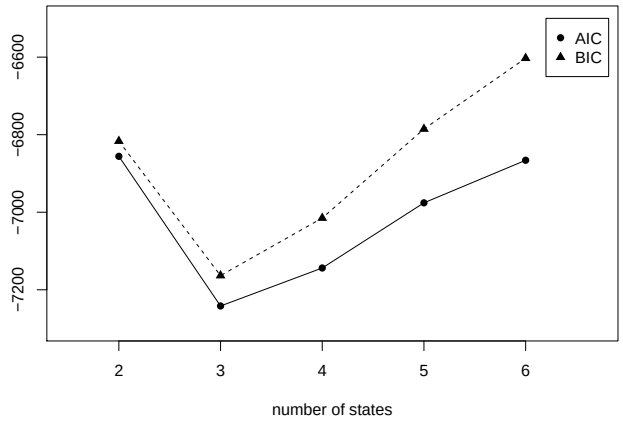

Figure 2: AIC and BIC for the HMM on the simulated data for $N = 2, \ldots, 5$ states.

Parameter	MSE (SD)
a_{11}	$3.71 \cdot 10^{-2}$ $(9.23 \cdot 10^{-3})$
a_{12}	$8.30 \cdot 10^{-3}$ $(1.17 \cdot 10^{-2})$
a_{13}	$8.01 \cdot 10^{-2}$ $(4.10 \cdot 10^{-2})$
a_{21}	$2.89 \cdot 10^{-2}$ $(2.32 \cdot 10^{-3})$
a_{22}	$2.07 \cdot 10^{-3}$ $(1.70 \cdot 10^{-3})$
a_{23}	$9.72 \cdot 10^{-4}$ $(1.69 \cdot 10^{-3})$
a_{31}	$1.31 \cdot 10^{-3}$ $(9.29 \cdot 10^{-3})$
a_{32}	$7.10 \cdot 10^{-8}$ $(3.54 \cdot 10^{-7})$
a_{33}	$1.32 \cdot 10^{-3}$ $(9.31 \cdot 10^{-3})$
ν_1	$2.00 \cdot 10^{-2}$ $(1.41 \cdot 10^{-1})$
ν_2	$2.00 \cdot 10^{-2}$ $(1.41 \cdot 10^{-1})$
ν_3	$< 2 \cdot 10^{-16} (< 2 \cdot 10^{-16})$

Table 1: MSE (SD) of the HMM parameters along 100 replications of the Baum-Welch algorithm with $N = 3$ states for the HMM.

	Viterbi Algorithm	k-means algorithm
CCR	0.857	0.591
MSE($\boldsymbol{\mu}_1; \widehat{\boldsymbol{\mu}}_1$) (SD)	0.085 (0.132)	0.131 (0.286)
MSE($\boldsymbol{\mu}_2; \widehat{\boldsymbol{\mu}}_2$) (SD)	0.636 (0.174)	0.890 (0.279)
MSE($\boldsymbol{\mu}_3; \widehat{\boldsymbol{\mu}}_3$) (SD)	0.953 (0.019)	1.051 (0.059)

Table 2: C.C.R. and MSE (S.D.) between the real and estimated means along 100 replications of the Viterbi and k-means algorithm.

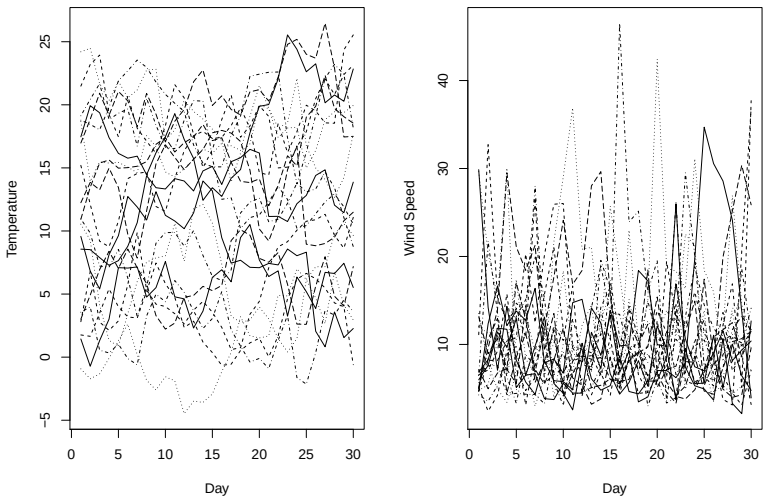

Figure 3: Monthly temperature and wind speed for the city of Basel.

4 Case study: Weather data

In this last part, we apply the described model to a real dataset regarding the weather in Basel, Switzerland; the dataset was obtained from [10]. In particular, our data consists of daily registrations of temperature and wind speed from 2008 to 2018. We consider each month as an observation of a statistical unit, in order to have 12 multivariate functional observations for every year, see Fig. 3. First, we apply our algorithm to the weather data with $N = 2, \ldots, 6$ states and compute every time the AIC and BIC of the model, as in (3.1); the results are showed in Fig. 4. We assume $N = 3$ to represent the optimal number of states, since both criteria exhibit the lowest value.

After choosing the number of states, we continue our analysis by applying the Baum-Welch and Viterbi algorithms. In particular, from the parameter estimation algorithm we obtain the following results:

$$\nu = (1, 0, 0) \qquad A = \begin{pmatrix} 0.708 & 0.292 & 0.000 \\ 0.079 & 0.498 & 0.423 \\ 0.000 & 0.181 & 0.819 \end{pmatrix},$$

i.e., the initial probabilities vector, the transition matrix and the functional parameters of the states for temperature and wind speed, which can be seen in Fig. 5. The three states are denoted by the Blue, Green and Red color, respectively. Since each statistical unit starts being observed during January, the vector ν of the initial probabilities only take probability 1 on the first state, which is the state with the lowest temperature and highest wind speed, representing the colder months. Moreover, the transition matrix shows how state 1 and 3

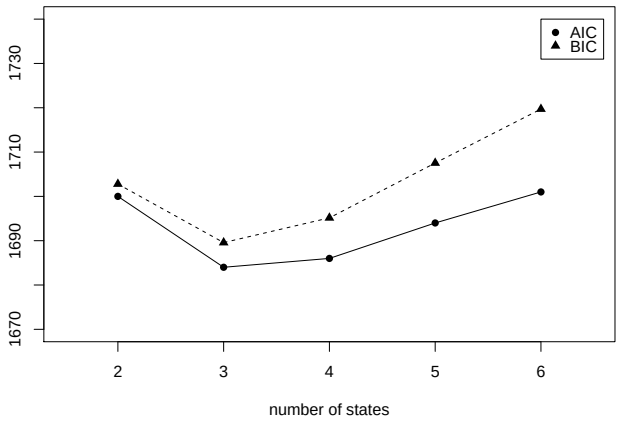

Figure 4: AIC and BIC for the weather data for $N = 2, \ldots, 6$ states.

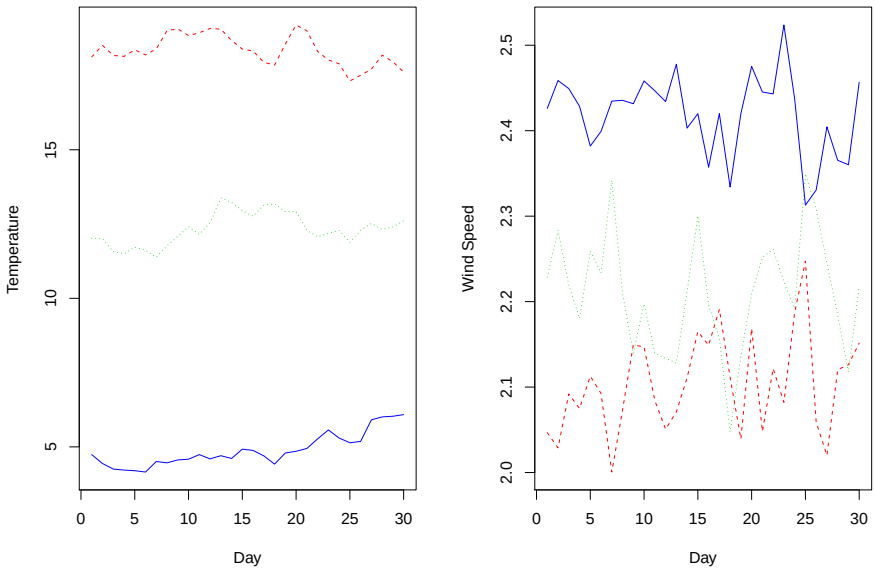

Figure 5: Plots of the functional parameters for the 3 states of temperature and precipitation.

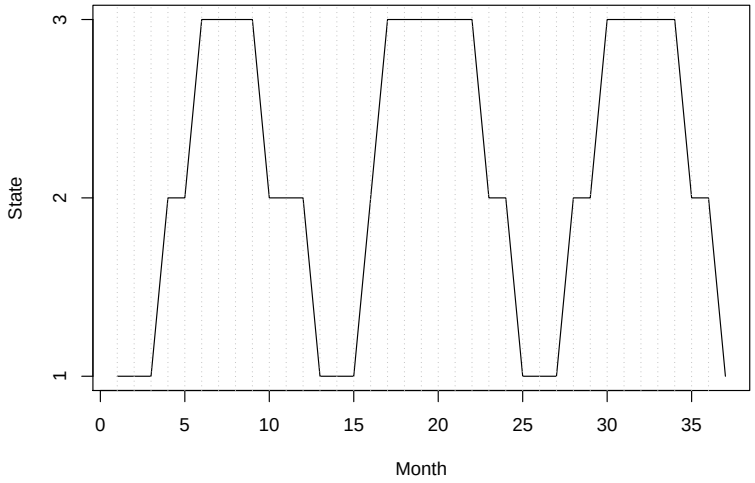

Figure 6: Labels of three years of observations obtained by the Viterbi algorithm.

	1	2	3
Number of months	40	35	57
Proportion per year	0.303	0.265	0.432

Table 3: Number of months for 11 years and proportion of months per year for a state of the HMM.

are the ones with higher probabilities for the model to remain in the state while state 2, representing the "mid-seasons", is the most unstable state and can be basically considered as a transition state between the other two.

After running the Viterbi algorithm, we obtained a set of labels for the observations; we show a subset for 3 years in Fig. 6. From the plot, we clearly notice the seasonality trend; moreover, it is clear how the results we obtained from the transition matrix are strengthened. The states corresponding to Summer and Winter are the longest and most stable, while the mid-seasons can be grouped together and are usually the shortest. In fact, as we can see in Tab. 3, the state linked to both mid-seasons is the less visited by the HMM over the period of time considered by the dataset.

5 Discussion and future developments

In this work, we faced the problem of estimating the parameters of a HMM where the output is a multivariate random curve. Then, using the Viterbi algorithm, we also considered and solved a classification problem. In particular, given a sequence of multivariate curves,

the HMM is able to detect the underlying structure of the time series system, by providing robust results; moreover, we noticed how sometimes, to perform a classification analysis, the shape of curves is not enough and by looking at the time order of the system we obtain better results.

As future development, it will be interesting to investigate the performance of the algorithm by extending the distance we used to more complex spaces, such as the Sobolev space H^1, and seeing how adding the information on the derivative of the functional data would affect the results.

References

[1] L. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a markov process. *Inequalities*, 3:1–8, 1972.

[2] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

[3] J. A. Bilmes et al. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126, 1998.

[4] O. Cappe, E. Moulines, and T. Ryden. *Inference in Hidden Markov Models*. 01 2005.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[6] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[7] F. Ferraty and P. Vieu. *Nonparametric functional data analysis: theory and practice*. Springer Science & Business Media, 2006.

[8] L. Horváth and P. Kokoszka. *Inference for functional data with applications*, volume 200. Springer Science & Business Media, 2012.

[9] A. Martino, G. Guatteri, and A. M. Paganoni. Multivariate hidden markov models for disease progression. *Mox Report 59/2018*, 2018.

[10] Meteoblue. Weather houston country club. *Meteoblue AG, www.meteoblue.com/*, 20 May 2019.

[11] L. J. Paas, J. K. Vermunt, and T. H. Bijmolt. Discrete time, discrete state latent markov modelling for assessing and predicting household acquisitions of financial

products. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(4):955–974, 2007.

[12] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.

[13] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[14] J. O. Ramsay. Functional data analysis. *Encyclopedia of Statistical Sciences*, 4, 2004.

[15] J. O. Ramsay and B. W. Silverman. *Applied functional data analysis: methods and case studies*. Springer, 2007.

[16] N. Tarabelloni, A. Arribas-Gil, F. Ieva, A. M. Paganoni, and J. Romo. *roahd: Robust Analysis of High Dimensional Data*, 2018. R package version 1.4.1.

[17] T. Tarpey and K. K. Kinateder. Clustering functional data. *Journal of classification*, 20(1):093–114, 2003.

[18] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967.

[19] L. R. Welch. Hidden markov models and the baum-welch algorithm. *IEEE Information Theory Society Newsletter*, 53(4):10–13, 2003.

[20] W. Zucchini, I. L. MacDonald, and R. Langrock. *Hidden Markov models for time series: an introduction using R*. Chapman and Hall/CRC, 2016.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

20/2019 Martino, A.; Guatteri, G.; Paganoni, A.M.
Hidden Markov Models for multivariate functional data

18/2019 Delpopolo Carciopolo, L.; Cusini, M.; Formaggia, L.; Hajibeygi, H.
Algebraic dynamic multilevel method with local time-stepping (ADM-LTS) for sequentially coupled porous media flow simulation

19/2019 Torti, A.; Pini, A.; Vantini, S.
Modelling time-varying mobility flows using function-on-function regression: analysis of a bike sharing system in the city of Milan.

17/2019 Antonietti,P.F.; De Ponti, J.; Formaggia, L.; Scotti, A.
Preconditioning techniques for the numerical solution of flow in fractured porous media

15/2019 Brandes Costa Barbosa, Y. A.; Perotto, S.
Hierarchically reduced models for the Stokes problem in patient-specific artery segments

14/2019 Antonietti, P.F.; Facciolà, C; Verani, M.
Mixed-primal Discontinuous Galerkin approximation of flows in fractured porous media on polygonal and polyhedral grids

16/2019 Antonietti, P.F.; Houston, P.; Pennesi, G.; Suli, E.
An agglomeration-based massively parallel non-overlapping additive Schwarz preconditioner for high-order discontinuous Galerkin methods on polytopic grids

13/2019 Manzoni, A; Quarteroni, A.; Salsa, S.
A saddle point approach to an optimal boundary control problem for steady Navier-Stokes equations

09/2019 Antonietti, P.F.; Facciola', C.; Verani, M.
Unified analysis of Discontinuous Galerkin approximations of flows in fractured porous media on polygonal and polyhedral grids

12/2019 Capezza, C.; Lepore, A.; Menafoglio, A.; Palumbo, B.; Vantini, S.
Control charts for monitoring ship operating conditions and CO2 emissions based on scalar-on-function regression