Functional Data Clustering via Piecewise Constant Nonparametric Density Estimation

Marc Boullé

Orange Labs, 2 avenue Pierre Marzin, 22300 Lannion, France

Abstract

In this paper, we present a novel way of analyzing and summarizing a collection of curves, based on piecewise constant density estimation. The curves are partitioned into clusters, and the dimensions of the curves points are discretized into intervals. The cross-product of these univariate partitions forms a data grid of cells, which represents a nonparametric estimator of the joint density of the curves and point dimensions. The best model is selected using a Bayesian model selection approach and retrieved using combinatorial optimization algorithms. The proposed method requires no parameter setting and makes no assumption regarding the curves; beyond functional data, it can be applied to distributional data. The practical interest of the approach for functional data and distributional data exploratory analysis is presented on two real world datasets.

Keywords: Functional data, Distributional data, Exploratory analysis, Clustering, Bayesianism, Model Selection, Density estimation

1. Introduction

Functional data analysis [1, 2, 3] relates to data samples where each observation is described by a function or curve, represented by a variable-length set of measure vectors (points). Functional data arise in many domains, such as measurements of the heights of children over a wide range of ages, daily records of precipitation at a weather station or hardware monitoring where each curve is a time series related to a physical quantity recorded at a specified sampling rate. Most statistical techniques designed for scalar data have their functional counterpart, including descriptive statistics, principal component analysis, supervised classification. In this paper, we focus on functional data exploratory analysis.

One of the key problems with functional data is that of data representation, with a preprocessing task of representing the curves by a fixed set of parameters or proposing a similarity between curves. Fixed size instances*variables representation allows to exploit most standard statistical techniques, whereas similarity provides the basis for clustering methods such as K-means (exploited for example in [4, 5, 6]). This problem has been studied for functional data as well as for time series. A standard approach is to approximate a function using a linear combination of basis functions, such as Fourier series [7], discrete wavelet transform [8], low degree polynomial functions [9] or spline basis functions [10, 11, 12]. In [13, 14], a hidden Markov model (HMM) is exploited as a parametric model of sequential data, and provides a similarity matrix according to the log-likelihood between sequence models and sequences. This similarity matrix is then used to build clusters of sequences, where each cluster is itself represented by a HMM. In [15], the self-organizing map (SOM) clustering algorithm is applied to functional data equipped with a similarity matrix. In [16], both the problem of segmentation of the curves (e.g. piecewise constant or linear) and clustering (K-means or SOM) are treated simultaneously. These approaches require both fixing some function parameters, such

Email address: email marc.boulle@orange.com (Marc Boullé)
URL: http://perso.rd.francetelecom.fr/boulle/ (Marc Boullé)

as polynomial degrees, the number of basis functions to use, number of segments for the representation of curves and setting the number of clusters for the clustering algorithm.

Nonparametric approaches have also been proposed, to better account for the potentially infinitely dimensional models behind functional data. In [17, 18], the functional data $Y = f(X) + \epsilon$ is summarized using nonparametric regression techniques, with a focus on the conditional mode, median and quantiles. Kernel techniques are employed that mainly locally weight the data using smoothing parameters. In [19, 20], the problem of density estimation of a random function is considered, by representing a function in the space of the eigenfunctions of principal component analysis. This kind of analysis reveals new patterns in functional data analysis, such as curves representing the mean or the mode in a curve dataset. Finally, clustering functional data is related to the problem of clustering time series data. In the survey [21], this problem is decomposed in three steps: choice of representation, with features extracted directly or indirectly from raw data or from models built from the raw data, choice of a similarity measure and choice of a clustering method.

In this paper, we propose a novel exploratory method for functional data, based on data grid models [22]. The collection of curves is represented by a fixed size dataset where each observation corresponds to a point of a curve with one categorical variable that stores the curve identifier and a finite dimensional numerical vector for the point variables. The categorical variable is partitioned into groups of curves and each numerical variable is discretized into intervals. The cross-product of these univariate partitions forms a multivariate partition, called data grid. By counting the frequencies in the multivariate parts (called cells) of this data grid, we obtain a nonparametric estimator of the joint density of all the variables [22]. A model selection technique based on a Bayesian approach with data dependent prior is applied to obtain an exact evaluation criterion for the posterior probability of joint density estimation data grid models. The prior is data dependent in that it exploits the number m of points and aims at modeling the data sample directly with a discrete distribution, not the true continuous-valued probability distribution. The best model is retrieved using combinatorial optimization algorithms, with a super-linear algorithmic complexity w.r.t. the number of points. In the case of functional data, grouping the values of the "curve identifier" variable can be interpreted as partitioning the curves into clusters, and discretizing each point variable provides an insightful summary of the curves, with an estimation of the joint density of the dimensions of each curve.

Compared to existing approaches, the benefit of our method is two-fold. It does not require any parameter, such as the choice of a family of basis functions, kernel parameters or a number of clusters, and it does not make any assumption regarding the curves such as their simplicity [16], smoothness as in regularization [23, 24] or capacity as in learning theory [25]. Compared to time series clustering approaches, our method exploits a single framework for the three-fold problem of choice of representation, similarity and clustering algorithm. It extends the functional data settings and can be applied to any distributional data, revealing new insights that have not previously been considered.

The rest of the paper is organized as follows. In Section 2, we present the MODL[1] approach for data grid models and apply it to joint density estimation and clustering for functional data. We illustrate the approach in Section 3 and present experimental results on two real world datasets in Section 4, which show what kind of exploratory analysis can be performed. Finally, we give a summary in Section 5.

2. MODL Approach for Functional Data Clustering

In this section, we first summarize the principles of data grid models introduced in [22] in the data mining field for supervised and unsupervised data preparation and show how these models can be applied to the problem of functional data clustering. We then adapt the approach to the case of functional data and finally describe the optimization algorithm.

2.1. Data Grid Models for Data Preparation in Data Mining

Data mining is "the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" [26]. Most data mining techniques work on flat tabular data, with one instance per row and one variable, numerical or categorical, per column. Supervised data mining aims at predicting the value of one target variable given the other explanatory variables: the task is classification in case of a categorical target variable

[1] MODL stands for Minimum Optimized Description Length

and regression in case of a target numerical variable. Unsupervised learning aims at discovering clusters in the data, association rules between the variables or at modeling correlations or joint density.

Data grid models [27, 22] have been introduced for the data preparation phase of the data mining process [28], which is a key phase, both time consuming and critical for the quality of the results. They allow to automatically, rapidly and reliably evaluate the class conditional probability of any subset of variables in supervised learning and the joint probability in unsupervised learning. Data grid models are based on a partitioning of each variable into intervals in the numerical case and into groups of values in the categorical case. The cross-product of the univariate partitions forms a multivariate partition of the representation space into a set of cells. This multivariate partition, called data grid, is a piecewise constant nonparametric estimator of the conditional or joint probability. The best data grid is searched using a Bayesian model selection approach and efficient combinatorial algorithms.

2.2. Application to Functional Data: principle

Let C be a collection of n curves $c_i, 1 \leq i \leq n$. Each curve $c_i = (p_{ij})_{j=1}^{m_i}$ has m_i observed values, the curve points. Each point $p_{ij} = (p_{ij1}, \ldots, p_{ijd})$ is a vector of finite dimension d. In the rest of the paper, without loss of generality and to keep the notation simple, we focus on the case where $d = 2$ and use X and Y for the two point dimensions. We have $c_i = (x_{ij}, y_{ij})_{j=1}^{m_i}$.

Let us take an example, with two curves c_1 and c_2, drawn on Figure 1, sampled at equidistant values for $x \in [0, 1]$ from the function $y = 1$ for c_1 and from the function $y = cos(\pi x)$ for c_2. Our sample dataset consists of $n = 2$ curves with respectively $m_1 = 4$ and $m_2 = 5$ points:

- $c_1 : (0, 1), (\frac{1}{3}, 1), (\frac{2}{3}, 1), (1, 1)$
- $c_2 : (0, 1), (\frac{1}{4}, \frac{\sqrt{2}}{2}), (\frac{1}{2}, 0), (\frac{3}{4}, -\frac{\sqrt{2}}{2}), (1, -1)$

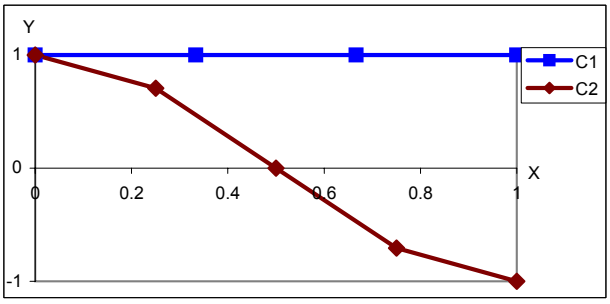

Figure 1: Two sample curves.

We propose to represent the collection of n curves as a unique dataset with three variables, C to store the curve identifier, X and Y for the point coordinates, and $m = \sum_{i=1}^{n} m_i$ observations. This is illustrated in Table 1.

Instead of considering a dataset of curves, where each instance has a variable-length description, we have a dataset of points represented in tabular format. We then can apply the data grid models in the unsupervised setting to estimate the joint density between the three variable $p(C, X, Y)$. The curve variable C is grouped into clusters of curves, whereas each point dimension X and Y is discretized into intervals. The cross-product of these univariate partitions forms a data grid of cells, which is piecewise constant per triplet of curve cluster, X interval and Y interval. As $p(X, Y|C) = \frac{p(C,X,Y)}{p(C)}$, this can also be interpreted as an estimator of the joint density between the point dimensions, which is constant per cluster of curves. This means that similar curves with respect to the joint density of their point dimensions will tend to be grouped into the same clusters. We formalize this in next section and illustrate it in Section 3.

C	X	Y
c_1	0	1
c_1	$\frac{1}{3}$	1
c_1	$\frac{2}{3}$	1
c_1	1	1
c_2	0	1
c_2	$\frac{1}{4}$	$\frac{\sqrt{2}}{2}$
c_2	$\frac{1}{2}$	0
c_2	$\frac{3}{4}$	$-\frac{\sqrt{2}}{2}$
c_2	1	-1

Table 1: Two curves represented as a unique dataset with three variables.

2.3. Density Estimation for Functional Data

We reformulate the data grid approach in the context of functional data clustering. A data grid provides a summary of a collection of curves with a piecewise constant joint density estimation of the curves and points. The finest representation is the data itself, with a data grid consisting of one cluster per curve and one interval per point value. The coarsest representation is obtained with a data grid consisting of one single cell containing all the points of all the curves, with the assumption that all the points are uniformly distributed in the $X * Y$ domain. The issue is to find a trade-off between the informativeness of the joint density estimation and its reliability, on the basis of the granularity of the data grid.

We introduce in Definition 1 a family of functional data clustering models, based on clusters of curves, intervals for each point dimension, and a multinomial distribution of all the points on the cells of the resulting data grid.

Definition 1. *A functional data clustering model is defined by:*

- *a number of clusters of curves,*
- *a number of intervals for each point dimension,*
- *the repartition of the curves into the clusters of curves,*
- *the distribution of the points of the functional dataset on the cells of the data grid,*
- *for each cluster of curves, the distribution of the points that belong to the cluster on the curves of the cluster.*

Notation.

- C: collection of curves
- $\mathcal{P}$: point dataset containing all points of C in tabular format
- C: curve variable
- X, Y: variables for the point dimensions
- $n = |C|$: number of curves
- $m = |\mathcal{P}|$: total number of points
- k_C: number of clusters of curves
- k_X, k_Y: number of intervals for variables X and Y
- $k = k_C k_X k_Y$: number of cells of the data grid
- $label_C(i)$: index of the cluster containing curve i
- n_{i_C}: number of curves in cluster i_C
- m_i: number of points for curve i

- m_{i_C}: cumulated number of points for curves of cluster i_C
- m_{j_X}: cumulated number of points for interval j_X of X
- m_{j_Y}: cumulated number of points for interval j_Y of Y
- $m_{i_C j_X j_Y}$: cumulated number of points for cell (i_C, j_X, j_Y) of the data grid

We assume that the numbers of curves n and points m are known in advance and we aim at modeling the joint distribution of the m points on the curve and the point dimensions.

Note. We do not assume that the numbers of points m_i per curve are all equal, neither that the points are ordered or at the same locations, nor that there is a smooth function underlying curve data such as $y_{ij} = x_{ij} + \epsilon_{ij}$ with errors ϵ_{ij}.

The family of models introduced in Definition 1 is completely defined by the parameters describing the partition of the curves into clusters

$$k_C, \{label_C(i)\}_{1 \leq i \leq n},$$

by the numbers of intervals for the point dimensions

$$k_X, k_Y,$$

by the parameters of the multinomial distribution of the points on the k cells of the data grid

$$\{m_{i_C j_X j_Y}\}_{1 \leq i_C \leq k_C, 1 \leq j_X \leq k_X, 1 \leq j_Y \leq k_Y},$$

and by the parameters of the multinomial distribution of the points belonging to each cluster of curves on the curves of the cluster

$$\{m_i\}_{1 \leq i \leq n}.$$

The numbers of curves per cluster n_{i_C} are derived from the partition of the curves into clusters: they do not belong to the model parameters. Similarly, the cumulated numbers of points per cluster of curves m_{i_C} or per intervals m_{i_X} and m_{i_Y} can be deduced by adding the frequencies of cells, according to

$$\begin{aligned}
m_{i_C} &= \sum_{j_X=1}^{k_X} \sum_{j_Y=1}^{k_Y} m_{i_C j_X j_Y}, \\
m_{j_X} &= \sum_{i_C=1}^{k_C} \sum_{j_Y=1}^{k_Y} m_{i_C j_X j_Y}, \\
m_{j_Y} &= \sum_{i_C=1}^{k_C} \sum_{j_X=1}^{k_X} m_{i_C j_X j_Y}.
\end{aligned}$$

It is noteworthy that the model parameters for the point dimensions exploit the ranks of the values in the dataset rather than the values themselves. Therefore, any model is invariant w.r.t. any monotonous transformation of the point dimensions and robust w.r.t. atypical values (outliers).

A functional data clustering model as defined in (1) can be seen as a generative model of the curve points, which corresponds to a piecewise constant join density estimation of the ranks of X and Y, per cluster of curves.

$$p(X \in \text{interval}_{j_X}, Y \in \text{interval}_{j_Y}, C \in \text{cluster}_{i_C}) = \frac{m_{i_C j_X j_Y}}{m}. \tag{1}$$

For a specific curve i belonging to a cluster i_C, we have

$$p(X \in \text{interval}_{j_X}, Y \in \text{interval}_{j_Y}, C = \text{curve}_i) = \frac{m_{i_C j_X j_Y}}{m} \frac{m_i}{m_{i_C}}. \tag{2}$$

In order to select the best model, we apply a Bayesian approach, where the best model is found by maximizing the probability $P(M|D)$ of the model given the data. Using Bayes rule and since the probability $P(D)$ is constant while varying the model, this is equivalent to maximizing $P(M)P(D|M)$, where $P(M)$ is the prior distribution on the model parameters and $P(D|M)$ is the likelihood. We choose the prior described in Definition 2, where the parameters for functional data clustering models are chosen hierarchically and uniformly at each level.

Definition 2. *The uniform hierarchical prior for the parameters of functional data clustering models is defined as follows:*

1. *the numbers of clusters k_C and of intervals k_X, k_Y are independent from each other, and uniformly distributed between* 1 *and n for the curves, between* 1 *and m for the point dimensions,*
2. *for a given number k_C of clusters, every partition of the n curves into k_C clusters are equiprobable,*
3. *for a model of size (k_C, k_X, k_Y), every distribution of the m points on the $k = k_C k_X k_Y$ cells of the data grid are equiprobable,*
4. *for a given cluster of curves, every distribution of the points in the cluster on the curves of the cluster are equiprobable,*
5. *for a given interval of X (resp. Y), every distribution of the ranks of the X (resp. Y) values of points are equiprobable.*

Using these assumptions and taking the negative log of the probabilities, this provides the evaluation criterion given in Theorem 3, which specializes to functional data clustering the unsupervised data grid model general criterion [29].

Theorem 3 (Evaluation criterion)**.** *A functional data clustering model M distributed according to a uniform hierarchical prior is Bayes optimal if the value of the following criteria is minimal*

$$\begin{aligned} c(M) = {} & \log n + 2\log m + \log B(n, k_C) \\ & + \log \binom{m+k-1}{k-1} + \sum_{i_C=1}^{k_C} \log \binom{m_{i_C} + n_{i_C} - 1}{n_{i_C} - 1} \\ & + \log m! - \sum_{i_C=1}^{k_C} \sum_{j_X=1}^{k_X} \sum_{j_Y=1}^{k_Y} \log m_{i_C j_X j_Y}! \\ & + \sum_{i_C=1}^{k_C} \log m_{i_C}! - \sum_{i=1}^{n} \log m_i! + \sum_{j_X=1}^{k_X} \log m_{j_X}! + \sum_{j_Y=1}^{k_Y} \log m_{j_Y}! \end{aligned} \tag{3}$$

Proof. See Appendix A. □

$B(n, k)$ is the number of divisions of n elements into k subsets (with eventually empty subsets). When $n = k$, $B(n, k)$ is the Bell number. In the general case, $B(n, k)$ can be written as $B(n, k) = \sum_{i=1}^{k} S(n, i)$, where $S(n, i)$ is the Stirling number of the second kind [30], which stands for the number of ways of partitioning a set of n elements into i nonempty subsets.

As negative log of probabilities are no other than coding length [31], criterion 3 can be interpreted according to the minimum description length (MDL) approach [32], with the coding length of the model plus the coding length of the dataset $\mathcal{P}$ given the model. The first line in Formula 3 relates to the prior distribution of the numbers of cluster k_C and of intervals k_X and k_Y, and to the specification the partition of the curves into clusters. The second line represents the specification of the parameters of the multinomial distribution of the m points on the k cells of the data grid, followed by the specification of the multinomial distribution of the points of each cluster on the curves of the cluster. The third line stands for the likelihood of the distribution of the points on the cells, by the mean of a multinomial term. The last line corresponds to the likelihood of the distribution of the points of each cluster on the curves of the cluster, followed by the likelihood of the distribution of the ranks of the X values (resp. Y values) in each interval.

2.4. Asymptotic Properties of the Evaluation Criterion

We present in this section an asymptotic approximation of criterion 3 and provide theoretical insights on why the method may work well for functional data clustering.

Null model. Let us first introduce the null model $M_\emptyset$, with one single cluster of curves, one single interval per point dimension, and one single cell containing all the points. Applying Formula 3, the cost $c(M_\emptyset)$ of the null model (its value according to evaluation criterion 3) reduces to

$$\begin{aligned} c(M_\emptyset) =& \log n + 2\log m + \log \binom{m+n-1}{n-1} \\ &+ \log \frac{m!}{m_1!m_2!\dots m_n!} + 2\log m! \end{aligned} \tag{4}$$

which corresponds to the posterior probability of the multinomial model for the distribution of the m points on the n curves and the posterior probability of the ranking of the values of each point dimension. This means that the curves and the dimensions of the points are described independently.

Let C_M, C_X, C_Y be the discretized versions of variables C, X, Y given a functional data clustering model M.

$$\begin{aligned} C_M =& i_C \Leftrightarrow C \in Cluster_{i_C} \\ X_M =& j_X \Leftrightarrow X \in Interval_{j_X} \\ Y_M =& j_Y \Leftrightarrow Y \in Interval_{j_Y} \end{aligned}$$

These discrete variables are distributed marginally and jointly according to:

$$\begin{aligned} C_M :& \{p_{i_C} = \frac{m_{i_C}}{m}\} \\ X_M :& \{p_{j_X} = \frac{m_{j_X}}{m}\} \\ Y_M :& \{p_{j_Y} = \frac{m_{j_Y}}{m}\} \\ C_M, X_M, Y_M :& \{p_{i_C j_X j_Y} = \frac{m_{i_C j_X j_Y}}{m}\} \end{aligned}$$

We introduce the notion of contrast of a model in Definition 4.

Definition 4. *The contrast of a functional data clustering model M is defined as the difference between the joint entropy of the variables C_M, X_M, Y_M and the sum of their individual entropies.*

$$Contrast(M) = H(C_M, X_M, Y_M) - H(C_M) - H(X_M) - H(Y_M) \tag{5}$$

Let us notice the contrast is always less than 0 since the joint entropy of variables is less than or equal to the sum of their individual entropies [33]. This inequality is an equality if and only if the variables are statistically independent.

We present in Theorem 5 our main result regarding the approximation of criterion 3.

Theorem 5 (Approximation of the evaluation criterion)**.** *The evaluation criterion of a functional data clustering model M distributed according to a uniform hierarchical prior can be approximated according to:*

$$|(c(M) - c(M_\emptyset)) - a(k,m)| < b(k_C, k, n, m) \tag{6}$$

where

$$\begin{aligned} a(k,m) =& m\, Contrast(M) \\ &+ (k-1)\log(m+k) \end{aligned} \tag{7}$$

and

$$\begin{aligned} b(k_C,k,n,m) =& n(1 + |\log \frac{1}{k_C}(1+\frac{m}{n})|) \\ &+ (\frac{1}{2}\log m + 2)(k+2) + k\log k. \end{aligned} \tag{8}$$

Proof. See Appendix B.
Mainly, the proof relies on a precise asymptotic approximation of $\log n!$ at the order $O(\frac{1}{n})$ and exploits Jensen's inequality. The *Contrast* term comes from the likelihood terms in criterion 3, on the basis of the approximation of the log of multinomial terms by entropy terms. □

Assuming that the number n of curves is fixed, we now exploit this approximation to derive a list a properties that provide theoretical ground to the method.

Theorem 6. *For a functional data clustering model M of fixed granularity (k_C, k_X, k_Y), the normalized difference of costs $\frac{c(M)-c(M_\emptyset)}{m}$ asymptotically converges to the difference between the joint entropy of the discretized variables C_M, X_m, Y_M and the sum of their individual entropies.*

$$\lim_{m\to\infty} \frac{c(M) - c(M_\emptyset)}{m} = Contrast(M) \tag{9}$$

Theorem 6 states that functional data clustering model allow to asymptotically approximate the joint entropy of the point variables, using fine grained functional clustering models. As $H(X_M, Y_M|C_M) = H(C_M, X_M, Y_M) - H(C_M)$, this allows to approximate the joint entropy of the curves. In particular, clusters of curves are likely to have good cost $c(M)$ in case of correlated curves.

Theorem 7. *For a set of functional data clustering models $M(m)$ of nested granularities with the number of cells $k(m)$ such that*

$$\lim_{m\to\infty} k(m) = \infty \quad and \quad \lim_{m\to\infty} \frac{k(m)\log m}{m} = 0,$$

the normalized difference of costs $\frac{c(M)-c(M_\emptyset)}{m}$ can be decomposed into a goodness of fit term 10, a regularization term 11 and an approximation term 12.

$$\begin{aligned} \frac{c(M) - c(M_\emptyset)}{m} &= Contrast(M) && (10)\\ &\quad + \alpha_m \frac{k(m)\log m}{m} && (11)\\ &\quad + \frac{1}{m}(O(\log m) + O(k(m)\log k(m))) && (12) \end{aligned}$$

with $1/2 \leq \alpha_m \leq 3/2$.

Theorem 7 shows how the Bayesian selection approach works: coarse grained models bring a small regularization term whereas fined grained models better fit the data, leading to potentially smaller joint entropy in the goodness of fit term. Overall, the optimal model granularity results from a balance between robustness (coarse grained models) and good fit of the data (fine grained models).

In the case of statistically independent variables C, X, Y, the goodness of fit term is null whereas the regularization term grows with the granularity of the model: the null model will be the best one. In case of dependent variables, in particular in case of statistically significant clusters of curves, the goodness of fit term is strictly negative, such that an informative model with clusters will be asymptotically more probable than the null model.

Overall, the method is likely to be resilient to spurious clusters and to uncover fine grained clusters in case of correlated curves, owing to a precise approximation of the joint entropy of the curve variables C, X, Y.

2.5. Optimization Algorithm

Functional data clustering models are no other than data grid models [29] applied to the case of joint density estimation of the curve and dimensions of the points. The space of data grid models is so large that straightforward algorithms almost surely fail to obtain good solutions within a practicable computational time. Given that criterion 3 is optimal, the design of sophisticated optimization algorithms is both necessary and meaningful. Such algorithms are described in [29, 34]. They finely exploit the sparseness of the data grid and the additivity of the criterion, and allow a deep search in the model space with $O(m)$ memory complexity and $O(m\sqrt{m}\log m)$ time complexity.

In this section, we give an overview of the optimization algorithms which are fully detailed in [29], and rephrase them using the functional data terminology. The optimization of a data grid is a combinatorial problem. The number of possible partitions of n curves is equal to the Bell number $B(n) = \frac{1}{e}\sum_{k=1}^{\infty}\frac{k^n}{k!}$, and the number of discretizations of m values is equal to 2^m. Even with very simple models having only two clusters of curves and two intervals per dimension, the number of models amounts to $2^n m^2$. An exhaustive search through the whole space of models is unrealistic. We describe in Algorithm 1 a greedy bottom up merge heuristic (GBUM) which optimizes the model criterion 3. The method starts with a fine grained model, with many clusters of curves and many intervals per dimension, up to the maximum model M_{Max} with one curve per cluster and one value per interval. It considers all the merges between any clusters of curves or between adjacent intervals for the dimension variables, and performs the best merge if the criterion decreases after the merge. The process is reiterated until no further merge decreases the criterion.

Algorithm 1 Greedy Bottom Up Merge heuristic (GBUM)

Require: M {Initial solution}
Ensure: $M^*, c(M^*) \leq c(M)$ {Final solution with improved cost}
1: $M^* \leftarrow M$
2: **while** improved solution **do**
3: $M' \leftarrow M^*$
4: **for all** Merge m between two clusters of C or two intervals of a dimension variable **do**
5: $M^+ \leftarrow M^* + m$ {Consider merge m for model M^*}
6: **if** $c(M^+) < c(M')$ **then**
7: $M' \leftarrow M^+$
8: **end if**
9: **end for**
10: **if** $c(M') < c(M^*)$ **then**
11: $M^* \leftarrow M'$ {Improved solution}
12: **end if**
13: **end while**

Each evaluation of the criterion for a data grid model requires $O(nm^2)$ time, since the initial model contains up to nm^2 cells (see Formula (3)) in the case of the maximal model M_{Max}. Each step of the algorithm relies on $O(n^2)$ evaluations of merges of clusters of curves and $O(2m)$ evaluation of merges of intervals, and there are at most $O(n+2m)$ steps, since the model becomes equal to the null model $M_\emptyset$ once all the possible merges have been performed. Overall, the time complexity of the algorithm is $O(n^4m^2+n^3m^3+nm^4)$ using a straightforward implementation of the algorithm. However, the method can be optimized in $O(m\sqrt{m}\log m)$ time, as demonstrated in [29]. The optimized algorithm mainly exploits the sparseness of the data, the additivity of the criterion and starts from non-maximal models with pre- and post-optimization heuristics.

- Collection of curves represented with datasets of points are sparse. Although a data grid model may contain $O(nm^2)$ cells, at most m cells are non empty. Since the contribution of empty cells is null in the criterion 3, each evaluation of a data grid can be performed in $O(m)$ time owing to specific algorithmic data structures.

- The additivity of the criterion means that it can be decomposed on the hierarchy of the components of the models: variables (curve and dimensions), parts (cluster and intervals), cells. Using this additivity property, all the merges between adjacent parts can be evaluated in $O(m)$ time. Furthermore, when the best merge is performed, the only impacted merges that need to be reevaluated for the next optimization step are the merges that share points with the best merge. Since the dataset is sparse, the number of reevaluations of models is small on average.

- Finally, the algorithm starts from initial fine grained solutions containing at most $O(\sqrt{m})$ clusters. Specific preprocessing and postprocessing heuristics are exploited to locally improve the initial and final solutions of Algorithm 1 by moving curves across clusters and moving interval bounds. The post-optimization algorithms are applied alternatively to each variable (curve or one dimension), for a frozen partition of the other variables. This allows to keep a $O(m)$ memory complexity and to bound the time complexity by $O(m\sqrt{m}\log m)$.

Sophisticated algorithmic data structures and algorithms are necessary to exploit these optimization principles and guarantee a $O(m\sqrt{m}\log m)$ time complexity for initial solutions exploiting at most $O(\sqrt{m})$ clusters of curves.

The optimized version of the greedy heuristic is time efficient, but it may fall into a local optimum. This problem is tackled using the variable neighborhood search (VNS) meta-heuristic [35], which mainly benefits from multiple runs of the algorithms with different random initial solutions. The first initial solution is generated with $\min(n, \sqrt[3]{m})$ random clusters of curves and $\sqrt[3]{m}$ random X and Y intervals in order to obtain $O(m)$ cells. Next initial solutions are generated in the neighborhood of previous local optima by randomly splitting clusters or intervals into subparts. In practice, the main heuristic described in Algorithm 1, with its guaranteed time complexity, is used to find a good solution as quickly as possible. The VNS meta-heuristic is exploited to perform anytime optimization: the more you optimize, the better the solution. In the experiments conducted throughout the paper, the anytime heuristic was used and stopped after 10 rounds of initialization.

The optimization algorithms summarized above have been extensively evaluated in [34], using a large variety of artificial datasets, where the true data distribution is known. Overall, the method is both resilient to noise and able to detect complex fine grained patterns. It is able to approximate any data distribution, provided that there are enough instances in the train data sample.

3. Illustration on Artificial Data

Let us consider the two functions introduced in Section 2.2, on the domain of x values $[0, 1]$, with an additive white Gaussian noise $N(0, \sigma)$ and standard deviation $\sigma = 0.25$:

- $f_1 : y = 1 + N(0, 0.25)$,
- $f_2 : y = cos(\pi x) + N(0, 0.25)$.

The conditional density $d(y|x)$ of f_1 and f_2 is drawn on Figure 2.

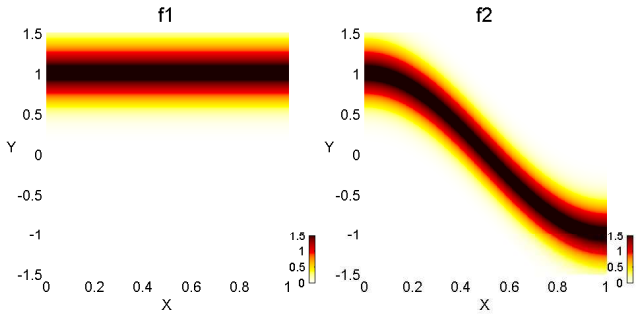

Figure 2: Two sample functions f_1 and f_2 drawn with their conditional density $d(y|x)$.

Let us consider a collection of 10 curves generated using function f_1 and 10 curves with function f_2. We generate a dataset $\mathcal{P}$ of 20000 points, on average 1000 per curve. Each point is a triple of values with a randomly chosen curve (among 20), a random x value (on domain $[0, 1]$) and a y value generated according to the function related to the curve.

We apply our functional data clustering method introduced in Section 2 on random subsets of $\mathcal{P}$ of increasing sizes. For very small sample sizes, there is not enough data to discover significant patterns, and our method produces one single cluster of curves, with one single interval for the X and Y variables. With only 5 points per curve on average, that is 100 points in the whole point dataset, our method recovers the underlying pattern and produces two clusters of 10 curves related to the f_1 and f_2 functions: the horizontal curves and the decreasing curves (cf. f_1 and f_2 in Figure 3). Our method is also a piecewise constant estimator of the joint probability $p(C, X, Y)$ of the three variables C, X, Y, based on both the clusters of curves and the discretization of the point dimensions X and Y. In our sample,

the C and X variables are both i.i.d. and independent. We thus have

$$\begin{aligned} p(Y|X,C) &= \frac{p(C,X,Y)}{p(X,C)}, \\ &= \frac{p(C,X,Y)}{p(X)p(C)}, \\ &\propto p(C,X,Y). \end{aligned}$$

For each cluster of curves, we have a piecewise constant estimation of the conditional probability $p(Y|C)$. Let us reuse the notation of Section 2.3, with $m_{i_C j_X j_Y}$ the number of points per cell (i_C, j_X, j_Y) of the data grid, and $m_{i_C j_X} = \sum_{j_Y=1}^{k_Y} m_{i_C j_X j_Y}$ the number of points per cluster i_C and interval j_X. We have

$$p(Y \in \text{interval}_{j_Y} | X \in \text{interval}_{j_X}, C \in \text{cluster}_{i_C}) = \frac{m_{i_C j_X j_Y}}{m_{i_C j_X}}.$$

We divide these estimated conditional probabilities by the width of interval_{j_Y} to obtain conditional densities that we draw in Figure 3, on the same basis as the true conditional densities pictured in Figure 2.

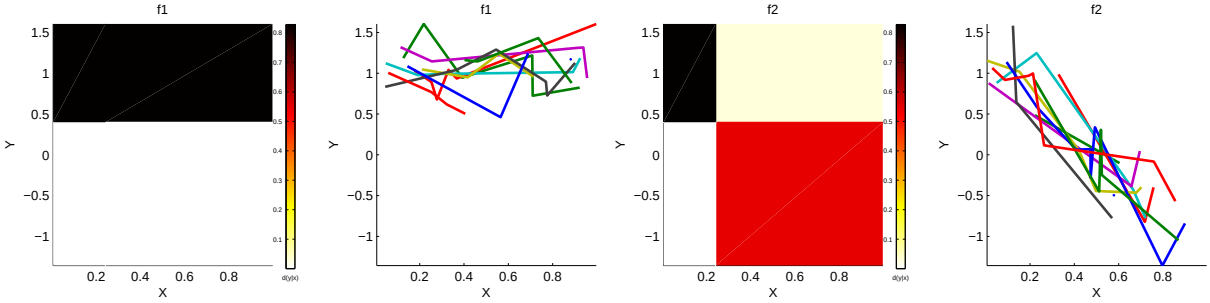

Figure 3: Estimation of the conditional density $d(y|x)$ and discovered clusters of curves, with 5 points per curve.

These estimations are very raw, with only two intervals for the X and Y variables, but they are obtained with only 100 points (5 points per curve) and provide a good summary of the underlying pattern: horizontal versus decreasing conditional density.

When our method is applied on a dataset of larger size, it still perfectly recovers the two cluster of curves and provides a refined version of the estimated conditional densities. With 1000 points per curve on average, that is 20000 points in the whole point dataset, the conditional density estimator exploits a joint discretization of the X, Y variables with 9 intervals for X and 12 for Y. This estimation, drawn in Figure 4, is a good approximation of the true conditional densities pictured in Figure 2.

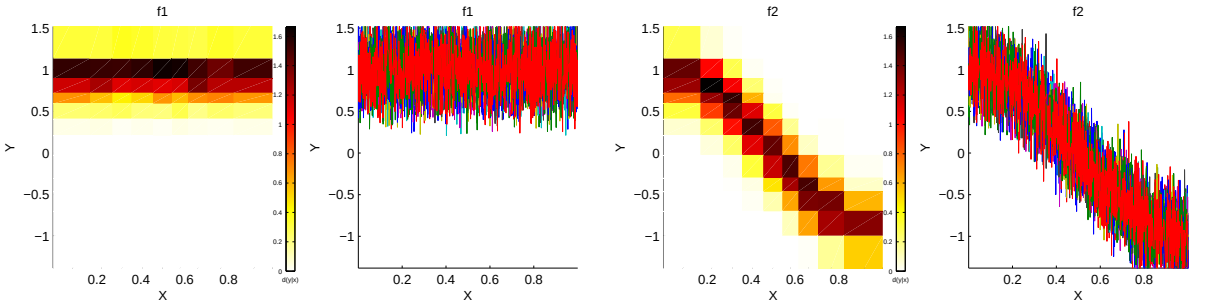

Figure 4: Estimation of the conditional density $d(y|x)$ and discovered clusters of curves, with 1000 points per curve.

We also performed the same experiment with the subset of 10 curves related to the function $f_1 : y = 1 + N(0, 0.25)$. Whatever the size of the dataset, the method always returns one single cluster, with one single interval for both the X and Y dimensions. The same experiment applied to f_2 curves still produces one single cluster of curves, but several intervals for the X and Y dimensions. This experimentally confirms the theoretical analysis presented in Section 2.4: in the case of f_1, the three variables C, X, Y are independent, such that the shortest way to encode the data is to encode each variable independently. One striking benefit of our approach is its robustness: it never produces spurious clusters whatever the sample size.

More results are available on a richer artificial dataset in [36], which demonstrate empirically that the method is able to approximate complex curve datasets provided that there is enough data, and never produces spurious clusters.

4. Experimental Results on Real Data

In this section, we apply the proposed approach on two real datasets and show what kind of exploratory analysis can be performed.

4.1. Topex/Poseidon Satellite

The first dataset[2] detailed in [37] consists of 472 waveforms registered by the Topex/Poseidon satellite around an area of 25 kilometers upon the Amazon river, with a variability originating from the differences in the ground type. Each waveform is a curve measured at 70 points. Figure 5 displays 10 curves chosen randomly from the dataset.

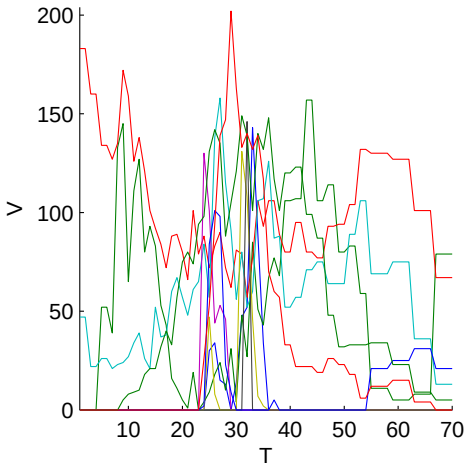

Figure 5: Topex/Poseidon dataset: 10 sample curves.

The original data comes in a format with one waveform per row, containing the 70 measures. We first reformat the data as a three-dimensional dataset, as described in Section 2.2. We obtain a point dataset of $472 * 70 = 33040$ points with one curve variable, the instance of waveform, and two dimension variables, the measure index ($T \in \{1, 2, \ldots, 70\}$) and the measured value ($V \in \{0, 1, \ldots, 255\}$). We apply the proposed method and obtain 33 clusters, summarized by the conditional probability $p(v|t)$ on a bivariate discretization $7 * 4$ of the measure dimensions.

Figure 6 displays a summary of all the clusters. They present a variety of forms, curves with one more or less heavy peak, with similar shapes but shifted peaks, flat noised curves with or without a stage.

[2]Dataset available at
http://www.math.univ-toulouse.fr/staph/npfda/npfda-datasets.html

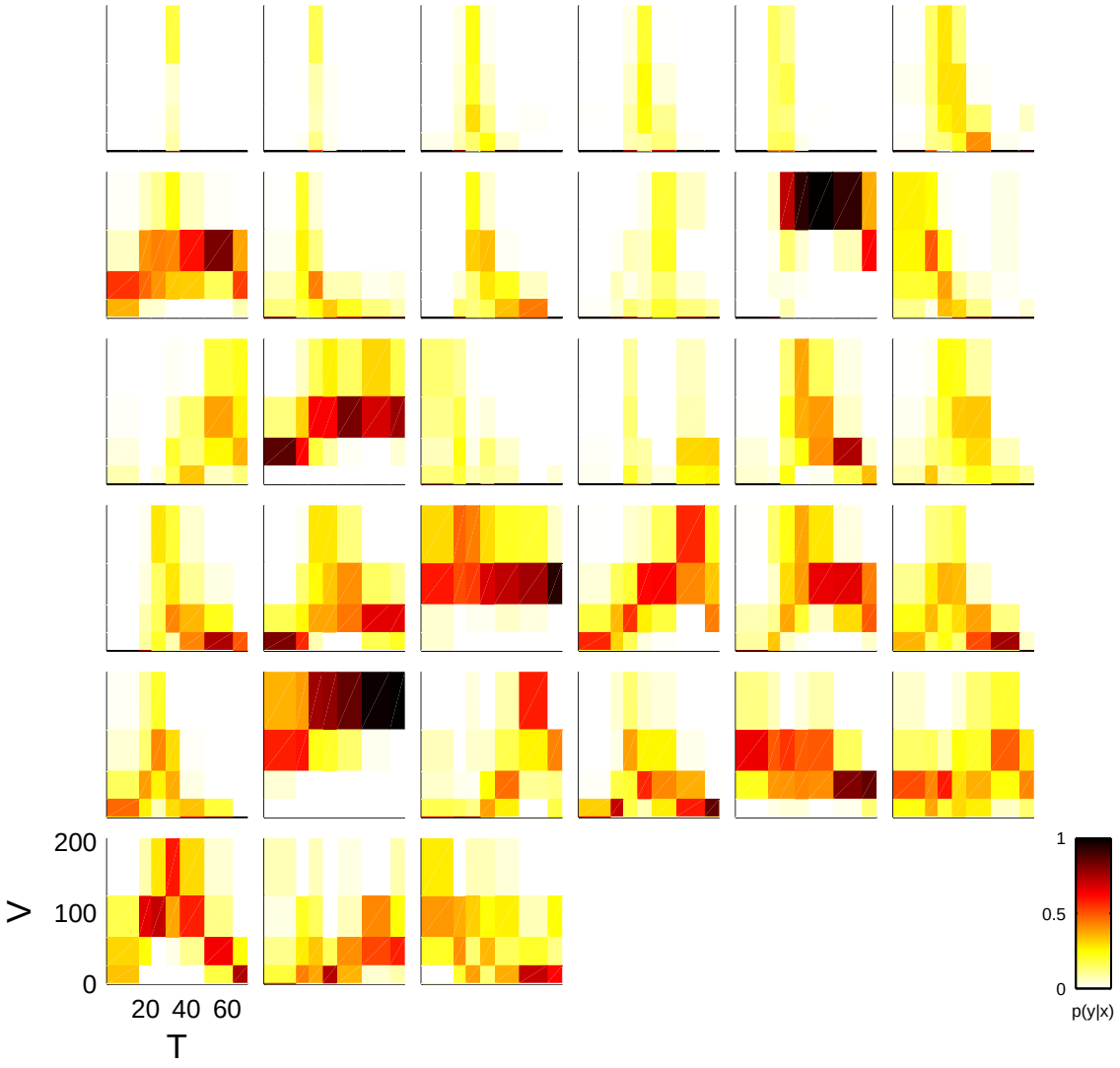

Figure 6: Topex/Poseidon dataset: all clusters.

Figure 7 focuses on four examples of clusters, to better illustrate the kind of summary provided by our approach. Each cluster is summarized according to the conditional probability

$$p(V \in \text{interval}_{j_V} | \; T \in \text{interval}_{j_T}, C \in \text{cluster}_{i_C}) = \frac{m_{i_C j_T j_V}}{m_{i_C j_T}}.$$

All the curves assigned to each cluster are also drawn. The figure shows the richness of the information retrieved in the probability-based summary of the clusters, as well the easy interpretation of the clusters. Although the individual curves are very noisy, the $7 * 4$ bivariate discretization grid provides a simple summary, with a good global fit of all the curves in a cluster. Nonparametric estimation of probability distribution goes far beyond the standard regression-based approaches, where the expectation of the curve target value only is estimated. The method not only summarizes the mean shape of the curves in a cluster, but also the variability of the curves inside each cluster, without any assumption

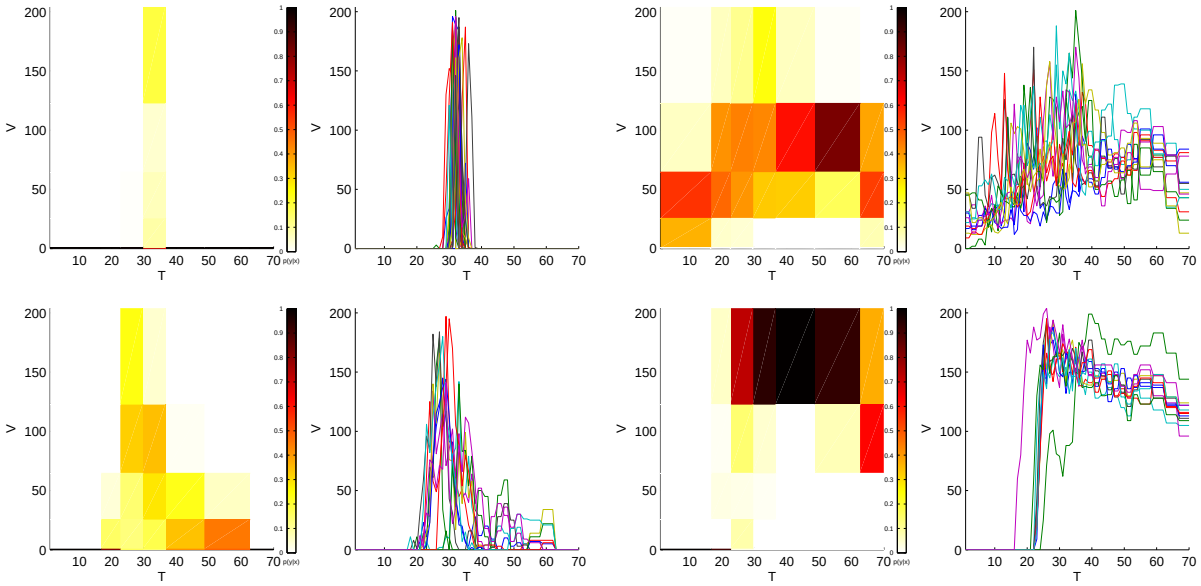

Figure 7: Topex/Poseidon dataset: four examples of clusters.

regarding constant Gaussian noise (like in ordinary least square regression technique) or any kind of parametric homoscedastic or heteroscedastic noise.

4.2. MNIST Handwritten Digits

The digit dataset[3] detailed in [38] consists of 8-bit gray scale images of "0" through "9" digits. The dataset was originally designed for the classification task of handwritten digit recognition, with a train set of 60 000 examples and a test set of 10 000 examples. Each image is a $28 * 28$ pixel box, with gray level from 0 to 255. We consider each image as the picture of a curve, and chose to keep the pixels with gray level above 128 as belonging to the curves. We exploit both the initial train and test sets and obtain a curve dataset related to handwritten digits, with 70 000 curves represented on a two-dimensional space X, Y, leading to a point dataset $\mathcal{P}$ containing about 7.2 million points. Figure 8 displays 100 curves chosen randomly from this curve dataset.

In this experiment, we apply our method as a exploratory analysis technique, and use the curve labels (digits) only in a second phase to evaluate the correlation between the clusters of curves and the curve labels. The interest of using this dataset with an exploratory analysis task is multiple.

- This is a large dataset, two orders of magnitude above the Topex/Poseidon satellite dataset. This provides a challenging benchmark to evaluate the scalability of an exploratory analysis technique.
- The curves are complex: they look closer to distribution of points than to functions. Any method assuming a functional relation between the point dimensions is likely to fail.
- Whereas this dataset has been extensively used for the classification task, it has received less attention for the task of exploratory analysis (see[39] for a deep learning approach). Many questions arise, related to the number and variety of "natural" patterns in this dataset, to the correlation between these patterns and the digits, to which digits exhibits a larger variety of shapes and whether they are harder to discriminate.
- As any educated people can be considered as an expert in handwritten digit recognition, this alleviates the evaluation of the understandability of the results of the proposed exploratory analysis.

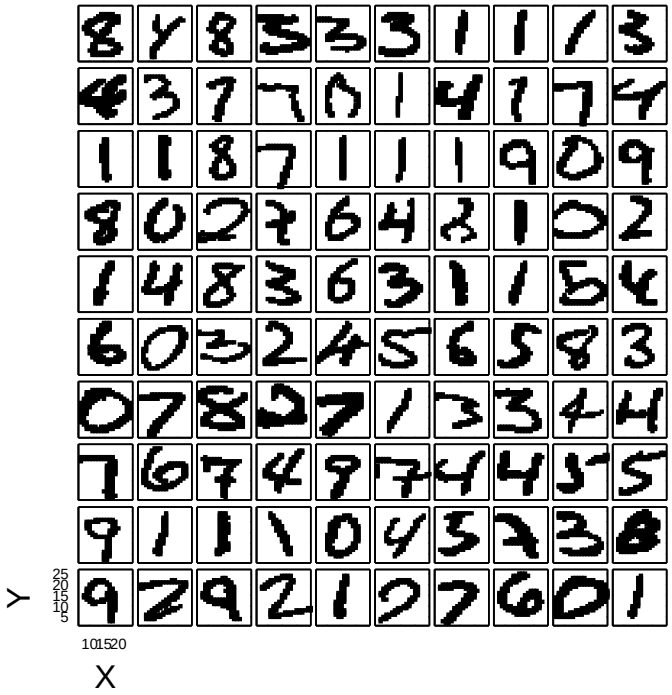

Figure 8: Handwritten digits datasets: 100 sample curves.

We apply the method presented in Section 2 and let the anytime heuristic run during one day, which corresponds to about ten rounds of optimization of the main algorithm. We obtain 568 clusters, summarized on a bivariate grid of size $15 * 21$. Since the curves are closer to point distributions rather than to functions, we focus on the the joint probability $p(x, y|c)$ per curve rather than on the conditional probability. Let us reuse the notation of Section 2.3, with $m_{i_C j_X j_Y}$ the number of points for cell (i_C, j_X, j_Y) of the data grid, and m_{i_C} the number of points for cluster i_C. We have

$$p(X \in \text{interval}_{j_X}, Y \in \text{interval}_{j_Y} | C \in \text{cluster}_{i_C}) = \frac{m_{i_C j_X j_Y}}{m_{i_C}}.$$

Figure 9 displays a summary of a subset of 100 randomly chosen clusters. Each cluster summary is a representation of a joint distribution, which highlights the dense regions in the bidimensional X, Y space. Interestingly, the shapes in the joint distribution space are very close to digits, and even more readable than the original digit curves, such as those presented in Figure 8.

Given this proximity of cluster summaries to digit shapes, we decide to reorganize the clusters according to their majority digit. Table 2 shows the number of clusters and the accuracy per digit. Figure 10 shows all the clusters for six among the 10 digits, sorted by decreasing frequency of their majority digit.

Digit "1" is the easiest, with only 35 clusters and an overall 98.0% of the curves assigned to the digit "1". The clusters are related to few shapes, but with a variety of orientation, thickness and (slight) curvature. Digits "4", "5" and "7" are a bit more complex, with 44, 52 and 47 clusters and a probability of correct assignment of 96.1%, 95.1% and 97.4%. Digits "8" and "9" are clearly the most difficult, with 78 and 46 clusters and a probability of correct assignment of 84.8% and 83.7%. Digit "8" exhibits the largest variety of shapes. Although it always consist of two loops, the variety comes from the thickness of the curve itself, the width and orientation of the overall shape and the

[3] Dataset available at http://yann.lecun.com/exdb/mnist

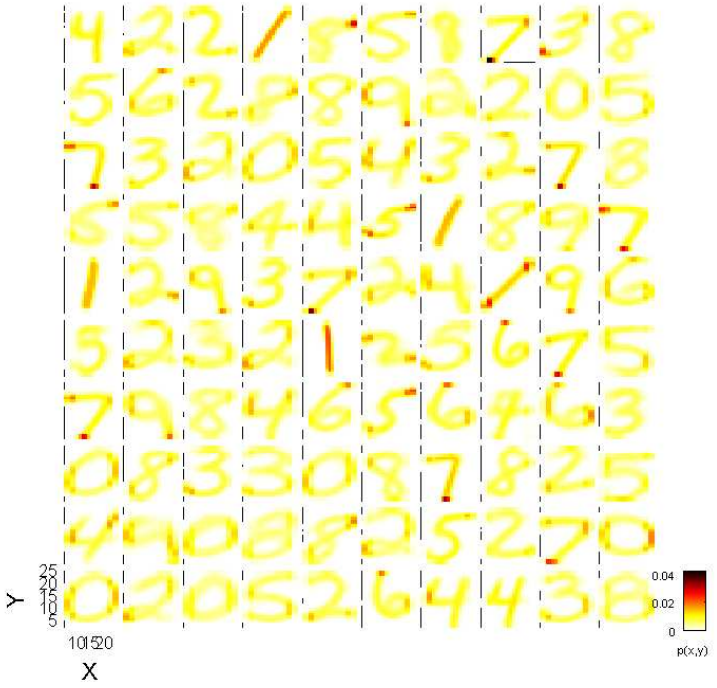

Figure 9: Handwritten digits datasets: 100 sample clusters with their $X * Y$ density.

Digit	Cluster number	Accuracy
0	65	96.7
1	35	98.0
2	76	96.4
3	74	94.5
4	44	96.1
5	52	95.1
6	51	96.9
7	47	97.4
8	78	84.8
9	46	83.7

Table 2: Handwritten digits datasets: number of clusters and accuracy per digit.

respective size of the upper and lower loop of the "8". It is noteworthy that in this representation space X, Y without any preprocessing, the "8" shapes are not always close to each other and do not even share many pixels. Constraining the clustering technique by a maximum number of clusters may have blurred the summaries and hidden potentially informative insights.

To study the correlation between the clusters and the digits, we sort the clusters by decreasing majority frequency and report the probability of each digit inside the clusters. The results are presented in Figure 11. Each column represent all the clusters with the same majority digit, sorted by decreasing frequency of this digit. Each row represent the distribution of a given cluster on the digits. For example, the narrowest column "C1" reports the distribution of digits in each of the 35 clusters assigned to "1", which almost contain 100% of digit "1" (see row "D1"). The largest

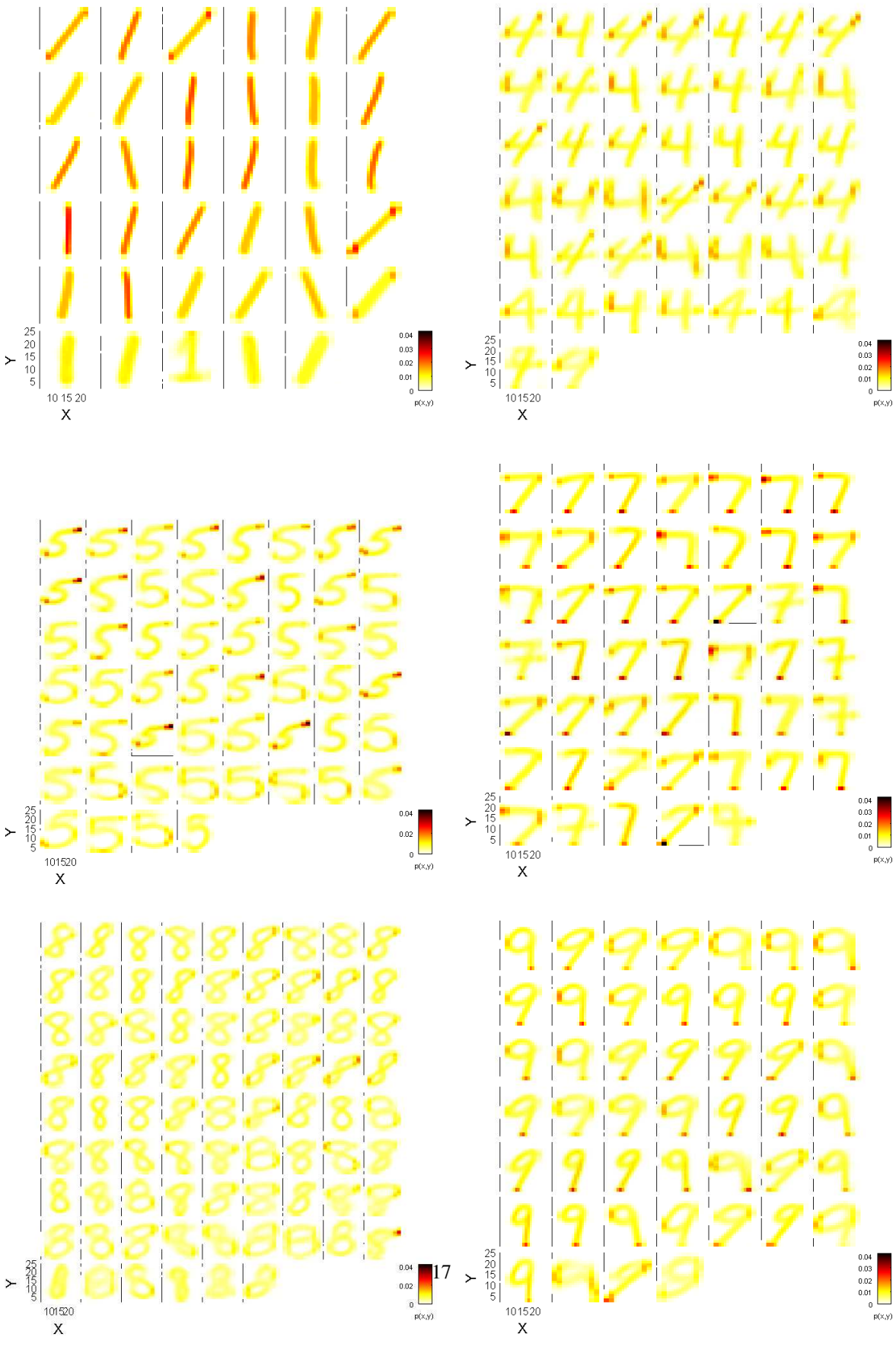

Figure 10: Handwritten digits datasets: clusters with their $X * Y$ density, organized according to their majority digit, for digits "1", "4", "5", "7", "8", "9".

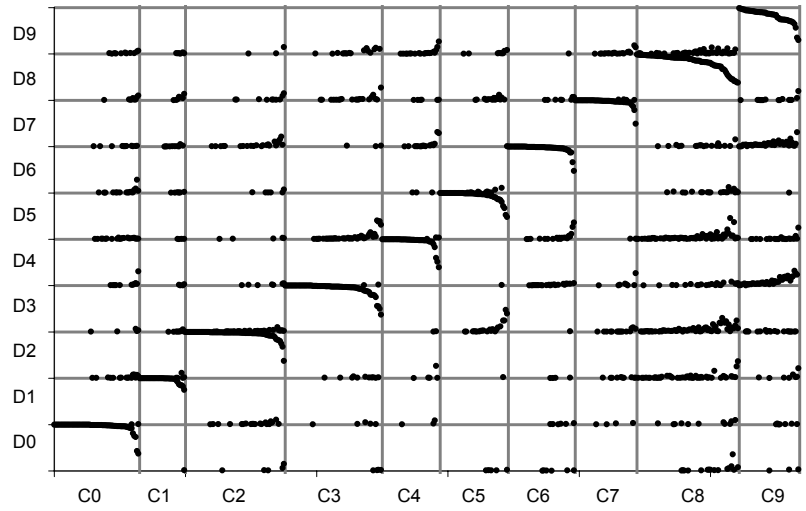

Figure 11: Handwritten digits datasets: distribution of digits per cluster.

column “C8” reports the distribution of digits in the 78 clusters assigned to “8”, which is a far more difficult digit. The last clusters have significant percentages of the other digits, with overall 0.9% of “0”, 1.9% of “2”, 4.5% of “3”, 3.7% of “5”, and 1.9% of “9”. The most difficult digit is “9”, but the the errors are mainly related to two other digits, with 8.6% of “4” and 5.5% of “7”.

Finally, we evaluate the clusters as a preprocessing method for digit classification, using a two-step method inspired by semi-supervised learning [40]. In a first step, all the train and test examples are used without the class labels (the digits) to train the unsupervised clustering model and produce the clusters. In a second step, each cluster is assigned to its majority class, using the train labels only. This class assignment is then used for prediction for the test examples. Using this protocol, where the test labels are ignored during the learning process, we obtain a test error rate of 5.9%.

Our reformatting of the initial handwritten digit dataset has kept only the pixels with gray level above 128, which results in some unknown information loss. Still, we can compare our test error rate with that of dedicated classification methods. This dataset has been widely used as a benchmark for tens of classification methods[4]. The test error rate of a basic linear classifier (one layer neural network) is 12.0% without preprocessing and 8.4% after deskewing. That of a k-nearest neighbors classifier with L2 norm and three neighbors is 5.0% without preprocessing and 2.4% with after deskewing. Support vector machines were also tried with Gaussian kernel or polynomial kernels with degree from 4 to 9; they obtained up to 0.8% error rate. These first results were reported in a vast comparative benchmark [38] and have since been improved with up to 0.4% error rate.

Our exploratory analysis method does not compete with the most sophisticated classifiers, but it performs remarkably well for a task it was not intended to and with an incomplete representation. Furthermore, it brings many informative insights w.r.t. the natural patterns in the dataset, which may suggest new preprocessing or normalization techniques to reduce the number of patterns and facilitate the task of digit recognition.

Overall, this last experiment on the MNIST handwritten digit dataset has demonstrated the scalability of our approach and its ability to process a complex curve dataset and discover potentially interesting patterns.

[4] See http://yann.lecun.com/exdb/mnist for a synthesis with many results and reference papers

5. Conclusion

In this paper, we have presented a new exploratory analysis method for functional data. Instead of considering a functional dataset as a data sample where the curves are the observations with variable-length representation, we chose to work with the fixed-size point dataset which instances are the curve points and variables are both one "curve identifier" categorical variable and the numerical point variables. The MODL approach based on data grid models introduced in [22] is applied to the case of functional datasets. By clustering the curves and discretizing each point variable, the method behaves as a nonparametric estimator of the joint density of both the curve and point variables. Experiments on two medium size and large real datasets show the benefits of the approach, which bring new insights in the exploratory analysis task, such as discovering the "natural" granularity of the point variables, the number of clusters, the estimation of the joint density in the point dimensions for each clusters of curves, with potentially multi-modal behavior, beyond the usual functional assumption.

Most alternative functional data analysis methods rely on strong assumptions, such as simple trends, smoothness, equally spaced observations, and require parameter tuning regarding the choice of the basis functions in the parametric case or the choice of the kernel parameters or the distance in the nonparametric case. On the contrary, our approach is both nonparametric, since it can can fit any functional data and even density data, and parameter-free, since no user parameter is required. The main originality of the modeling approach is that it is data dependent and non-asymptotic in essence: it aims at modeling the finite functional sample directly. The modeling task is then easier, with finite modeling space and model priors which essentially reduce to counting.

Interestingly, whereas the primary purpose of the method is density estimation, it comes with insightful byproducts such as the clustering of the curves and the discretization of the curve dimensions, which reduces the dimensionality of the curves in a fixed-size space. As the method automatically infers the optimal granularity of the clustering, it tends to build more and more clusters as the amount of data increases, up to potentially one cluster per curve. In future work, we plan to derive a similarity from the model evaluation criterion and organize the clusters into a hierarchy in order to alleviate the exploratory analysis task.

Appendix A. Evaluation Criterion

Proof of Theorem 3 (Evaluation criterion).

Let M be a functional data clustering model distributed according to the uniform hierarchical prior (2).

The Bayes optimal model is found by maximizing $P(M)P(D|M)$.

We have $M = \{k_C, k_X, k_Y, \{label_C(i)\}_i, \{m_{i_C j_X j_Y}\}_{i_C, j_X, j_Y}, \{m_i\}_i\}$.

Let us firt evaluate the prior probability of M.

$$\begin{aligned} P(M) =& P(k_C, k_X, k_Y) \\ & P(\{label_C(i)\}_i | k_C, k_X, k_Y) \\ & P(\{m_{i_C j_X j_Y}\}_{i_C, j_X, j_Y} | \{label_C(i)\}_i, k_C, k_X, k_Y) \\ & P(\{m_i\}_i | \{m_{i_C j_X j_Y}\}_{i_C, j_X, j_Y}, \{label_C(i)\}_i, k_C, k_X, k_Y) \end{aligned}$$

The first hypothesis introduced in Definition (2) gives that

$$P(k_C, k_X, k_Y) = \frac{1}{n}\frac{1}{m}\frac{1}{m}.$$

From the second prior hypothesis, since $B(n, k_C)$ is the number of partitions of n curves into k_C clusters, we get

$$P(\{label_C(i)\}_i | k_C, k_X, k_Y) = \frac{1}{B(n, k_C)}.$$

Computing the number of parameters of the multinomial distribution of m points on $k = k_C k_X k_Y$ cells is a combinatorics problem, the solution of which is $\binom{m+k-1}{k-1}$. Using the third hypothesis of prior (2), we obtain

$$P(\{m_{i_C j_X j_Y}\}_{i_C, j_X, j_Y} | \{label_C(i)\}_i, k_C, k_X, k_Y) = \frac{1}{\binom{m+k-1}{k-1}}.$$

Finally, in each cluster containing the m_{i_C} points of n_{i_C} curves, we have to compute the number of parameters of the multinomial distribution of m_{i_C} points on n_{i_C} curves. Using the fourth prior hypothesis, we get

$$P(\{m_i\}_i|\{m_{i_C j_X j_Y}\}_{i_C,j_X,j_Y}, \{label_C(i)\}_i, k_C, k_X, k_Y) = \prod_{i_C=1}^{k_C} \frac{1}{\binom{m_{i_C}+n_{i_C}-1}{n_{i_C}-1}}.$$

Overall, the prior probability of a model is

$$P(M) = \frac{1}{n}\frac{1}{m}\frac{1}{m}\frac{1}{B(n,k_C)}\frac{1}{\binom{m+k-1}{k-1}}\prod_{i_C=1}^{k_C}\frac{1}{\binom{m_{i_C}+n_{i_C}-1}{n_{i_C}-1}}.$$

The prior terms being explicited, it remains to evaluate the likelihood of the data given the model. We decompose the likelihood as $P(D|M) = P_1(D|M)P_2(D|M)P_3(D|M)$ where:

1. $P_1(D|M)$ is the probability of observing the points in each cell of the 3-dimensional data grid knowing the multinomial distribution of the m point on k cells,
2. $P_2(D|M)$ is the probability of observing the points of each cluster in the curves knowing the multinomial distributions of the m_{i_C} points of each cluster i_C on n_{i_C} curves,
3. $P_3(D|M)$ is the probability of observing the ranks of the points in each X or Y interval.

For a given set of multinomial parameters, the number of observable distinct permutations is given by the multinomial coefficient. Thus we get

$$P_1(D|M) = \frac{1}{\frac{m!}{\prod_{i_C=1}^{k_C}\prod_{j_X=1}^{k_X}\prod_{j_Y=1}^{k_Y} m_{i_C j_X j_Y}!}}$$

and

$$\begin{aligned} P_2(D|M) =& \frac{1}{\prod_{i_C=1}^{k_C}\frac{m_{i_C}!}{\prod_{i,label_C(i)=i_C}\log m_i!}} \\ =& \frac{1}{\frac{\prod_{i_C=1}^{k_C} m_{i_C}!}{\prod_{i=1}^{n}\log m_i!}}. \end{aligned}$$

From the last hypothesis in prior (2), every distribution of the ranks of the X (resp. Y) values of points in each interval are equiprobable, and thus

$$P_3(D|M) = \prod_{j_X=1}^{k_X}\frac{1}{m_{j_X}!}\prod_{j_Y=1}^{k_Y}\frac{1}{m_{j_Y}!}.$$

Taking the negative log of $P(M)P_1(D|M)P_2(D|M)P_3(D|M)$, we get

$$\begin{aligned} c(M) =& \log n + 2\log m + \log B(n,k_C) \\ &+ \log\binom{m+k-1}{k-1} + \sum_{i_C=1}^{k_C}\log\binom{m_{i_C}+n_{i_C}-1}{n_{i_C}-1} \\ &+ \log m! - \sum_{i_C=1}^{k_C}\sum_{j_X=1}^{k_X}\sum_{j_Y=1}^{k_Y}\log m_{i_C j_X j_Y}! \\ &+ \sum_{i_C=1}^{k_C}\log m_{i_C}! - \sum_{i=1}^{n}\log m_i! + \sum_{j_X=1}^{k_X}\log m_{j_X}! + \sum_{j_Y=1}^{k_Y}\log m_{j_Y}! \end{aligned}$$

The claim follows. □

Appendix B. Approximation of the Evaluation Criterion

We first introduce new notations to study the precise asymptotic behavior of functions, then present asymptotic approximations of some combinatorial functions, finally establish the asymptotic approximation of the evaluation criterion of functional data clustering models.

Appendix B.1. Notations for Precise Asymptotic Study

Let us first recall the big-O notation used to describe the limiting behavior of functions when the argument tends to a particular value or to infinity.

For example, let f(n) and g(n) be two functions defined on $\mathbb{N}$.

$$f(n) = O(g(n)) \text{ as } n \to \infty \;\Leftrightarrow \exists C \in \mathbb{R}^+, \exists n_0 \in \mathbb{N}, \forall n > n_0, |f(n)| < Cg(n).$$

In order to study the precise asymptotic behavior of a function, we extend the big-O notation using the following definitions.

Big-$O^{\pm}$ notation.

$$f(n) = O^{\pm}(g(n)) \text{ as } n \to \infty \;\Leftrightarrow \forall n \in \mathbb{N}, |f(n)| < g(n).$$

Big-O^{+} notation.

$$f(n) = O^{+}(g(n)) \text{ as } n \to \infty \;\Leftrightarrow \forall n \in \mathbb{N}, 0 \leq f(n) < g(n).$$

As an example, let us consider the Stirling's series for the logarithm of factorials.

$$\log n! = n\log n - n + \frac{1}{2}\log 2\pi n + \frac{1}{12n} - \frac{1}{360n^3} + \frac{1}{1260n^5} - \frac{1}{1680n^7} + \dots$$

This provides the usual asymptotic approximation

$$\log n! = n\log n - n + \frac{1}{2}\log 2\pi n + O(\frac{1}{n}).$$

Using the precise asymptotic notation, the following equalities allow more accurate approximations for $n \in \mathbb{N}^+$.

$$\begin{aligned} \log n! =& n\log n - n + \frac{1}{2}\log 2\pi n + O^{\pm}(\frac{1}{12n}) \\ \log n! =& n\log n - n + \frac{1}{2}\log 2\pi n + O^{+}(\frac{1}{12n}). \end{aligned} \tag{B.1}$$

Let us notice that $O^+(1/12n) = 1/12\ O^+(n) = 1/12n\ O^+(1)$.

Appendix B.2. Preliminary Precise Asymptotic Results

Let us present an asymptotic formula for the logarithm of binomial coefficients.

Lemma 8. *For $n \in \mathbb{N}^+, k \in \mathbb{N}^+, k < n$, we have*

$$\begin{aligned} \log\binom{n}{k} =& k(\log n - \log k) + k - \frac{1}{2}\log 2\pi k \\ & - kO^+(\frac{k}{n}) + O^+(\frac{1}{12n}) - O^+(\frac{1}{12k}) - O^+(\frac{1}{12(n-k)}) \end{aligned}$$

Proof. Applying the asymptotic approximation B.1 of logarithm of factorials, we get

$$\begin{aligned}
\log\binom{n}{k} =& \log n! - \ln k! - \log(n-k)!,\\
=& n\log n - n + \frac{1}{2}\log 2\pi n + O^+(\frac{1}{12n})\\
&- (k\log k - k + \frac{1}{2}\log 2\pi k) - O^+(\frac{1}{12k})\\
&- ((n-k)\log(n-k) - (n-k) + \frac{1}{2}\log 2\pi(n-k)) - O^+(\frac{1}{12(n-k)}),\\
=& n\log n - k\log k - (n-k)\log n(1-\frac{k}{n})\\
&+ \frac{1}{2}(\log 2\pi n - \log 2\pi k - \log 2\pi n(1-\frac{k}{n}))\\
&+ O^+(\frac{1}{12n}) - O^+(\frac{1}{12k}) - O^+(\frac{1}{12(n-k)}),\\
=& k(\log n - \log k) - \frac{1}{2}\log 2\pi k\\
&- (n-k+\frac{1}{2})\log(1-\frac{k}{n})\\
&+ O^+(\frac{1}{12n}) - O^+(\frac{1}{12k}) - O^+(\frac{1}{12(n-k)}).
\end{aligned}$$

Using the Taylor series for $\log(1+x)$, we have the following non asymptotic equality for $|x| < 1$:

$$\log = (1+x) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}\,O^+(x^4).$$

We obtain

$$\begin{aligned}
\log\binom{n}{k} =& k(\log n - \log k) - \frac{1}{2}\log 2\pi k\\
&- (n-k+\frac{1}{2})(-\frac{k}{n} - \frac{k^2}{2n^2} - \frac{k^3}{3n^3} - \frac{1}{4}\,O^+(\frac{k^4}{n^4}))\\
&+ O^+(\frac{1}{12n}) - O^+(\frac{1}{12k}) - O^+(\frac{1}{12(n-k)}),\\
=& k(\log n - \log k) + k - \frac{1}{2}\log 2\pi k\\
&- kO^+(\frac{k}{n}) + O^+(\frac{1}{12n}) - O^+(\frac{1}{12k}) - O^+(\frac{1}{12(n-k)})
\end{aligned}$$

□

A relaxed approximation of the logarithm of binomial coefficients is given in Lemma 9.

Lemma 9. *For $n \in \mathbb{N}^+, k \in \mathbb{N}^+, k < n$, we have*

$$\log\binom{n}{k} = k(\log n - \log k) + k - \frac{1}{2}\log 2\pi k - k\,O^+(1) + \frac{1}{6}\,O^{\pm}(1)$$

We now present an asymptotic formula for the logarithm of multinomial coefficients.

Lemma 10. *Let $n = \sum_{i=1}^{k} n_i, n \in \mathbb{N}^+, n_i \in \mathbb{N}$, $k^* = \sum_{n_i>0} 1$ and $H(p)$ be the entropy of the distribution defined by $\{p_i = \frac{n_i}{n}\}_i$. We have*

$$\log\frac{n!}{\prod_i^k n_i!} = nH(p) + \frac{1}{2}\log 2\pi n - (\frac{1}{2}\log 2\pi\frac{n}{k^*} + \frac{1}{12})\,O^+(k^*)$$

Proof. We have

$$\log \frac{n!}{\prod_{i=1}^{k} n_i!} = \log n! - \sum_{i=1}^{k} \log n_i!.$$

Applying the approximation B.1 of logarithm of factorials, we get

$$\begin{aligned}
\log \frac{n!}{\prod_i^k n_i!} =& n \log n - n + \frac{1}{2} \log 2\pi n + O^+(\frac{1}{12n}) \\
& - \sum_{n_i>0} (n_i \log n p_i - n_i + \frac{1}{2} \log 2\pi n_i + O^+(\frac{1}{12n_i})) \\
=& - n \sum_{i=1}^{k} p_i \log p_i + \frac{1}{2} \log 2\pi n - \frac{1}{2} \sum_{n_i>0} \log 2\pi n_i \\
& + O^+(\frac{1}{12n}) - \sum_{n_i>0} O^+(\frac{1}{12n_i})
\end{aligned}$$

We have $nH(p) = -n \sum_{i=1}^{k} p_i \log p_i$.
Since the logarithm function is concave, we can use Jensen's inequality and get

$$\frac{1}{k^*} \sum_{n_i>0} \log n_i \leq \log \frac{n}{k^*}.$$

Since $\forall n_i > 0, 1/n_i > 1/n$, we have

$$0 \leq -O^+(\frac{1}{12n}) + \sum_{n_i>0} O^+(\frac{1}{12n_i}) \leq \frac{1}{12}\ O^+(k^*).$$

Finally, we obtain

$$\log \frac{n!}{\prod_i^k n_i!} = nH(p) + \frac{1}{2} \log 2\pi n - (\frac{1}{2} \log 2\pi \frac{n}{k^*} + \frac{1}{12})\ O^+(k^*)$$ □

Appendix B.3. Main Result

Proof of Theorem 5 (Approximation of the evaluation criterion).

Let us calculate the difference of cost between a model M and the null model $M_\emptyset$.

$$\begin{aligned}
c(M) - c(M_\emptyset) =& \log B(n, k_C) \\
& + \log \binom{m+k-1}{k-1} - \log \binom{m+n-1}{n-1} + \sum_{i_C=1}^{k_C} \log \binom{m_{i_C} + n_{i_C} - 1}{n_{i_C} - 1} \\
& - \sum_{i_C=1}^{k_C} \sum_{j_X=1}^{k_X} \sum_{j_Y=1}^{k_Y} \log m_{i_C j_X j_Y}! \\
& + \sum_{i_C=1}^{k_C} \log m_{i_C}! + \sum_{j_X=1}^{k_X} \log m_{j_X}! + \sum_{j_Y=1}^{k_Y} \log m_{j_Y}! - 2 \log m!
\end{aligned}$$

Let us notice that

$$\log \binom{m+k-1}{k-1} = \log \binom{m+k}{k} - (\log(m+k) - \log(k)).$$

Using approximation 9, we have

$$\log \binom{m+k-1}{k-1} = (k-1)(\log(m+k) - \log k) + k - \frac{1}{2} \log 2\pi k - k\ O^+(1) + \frac{1}{6}\ O^\pm(1).$$

Using the preceding equality, the approximation 10 of multinomial coefficients and the bound $\log B(n, k_C) \leq n \log k_C$, we obtain

$$\begin{aligned}
c(M) - c(M_\emptyset) = {} & O^+(n \log k_C) \\
& + (k-1)(\log(m+k) - \log k) + k - \frac{1}{2}\log 2\pi k - k\, O^+(1) + \frac{1}{6}\, O^\pm(1) \\
& - (n-1)(\log(m+n) - \log n) - n + \frac{1}{2}\log 2\pi n + n\, O^+(1) + \frac{1}{6}\, O^\pm(1) \\
& + \sum_{i_C=1}^{k_C} ((n_{i_C} - 1)(\log(m_{i_C} + n_{i_C}) - \log n_{i_C}) + n_{i_C} - \frac{1}{2}\log 2\pi n_{i_C}) \\
& + \sum_{i_C=1}^{k_C} (-n_{i_C}\, O^+(1) + \frac{1}{6}\, O^\pm(1)) \\
& + mH(C_M, X_M, Y_M) + \frac{1}{2}\log 2\pi m - (\frac{1}{2}\log 2\pi \frac{m}{k^*} + \frac{1}{12})\, O^+(k^*) \\
& - mH(C_M) - \frac{1}{2}\log 2\pi m + (\frac{1}{2}\log 2\pi \frac{m}{k_C^*} + \frac{1}{12})\, O^+(k_C^*) \\
& - mH(X_M) - \frac{1}{2}\log 2\pi m + (\frac{1}{2}\log 2\pi \frac{m}{k_X^*} + \frac{1}{12})\, O^+(k_X^*) \\
& - mH(Y_M) - \frac{1}{2}\log 2\pi m + (\frac{1}{2}\log 2\pi \frac{m}{k_Y^*} + \frac{1}{12})\, O^+(k_Y^*)
\end{aligned}$$

$$\begin{aligned}
c(M) - c(M_\emptyset) = {} & O^+(n \log k_C) \\
& - (k - \frac{1}{2})\log k + O^+(k) \\
& + (n - \frac{1}{2})\log n + O^+(n) \\
& + (k-1)\log(m+k) - (n-1)\log(m+n) + \frac{1}{2}\, O^\pm(1) \\
& + O^+((n-k_C)\log(m+n)) - O^+((n-k_C)\log(n)) - \ O^+(k_C \log \sqrt{2\pi}) \\
& - O^+(n) + \frac{1}{6}\, O^\pm(k_C) \\
& + (\frac{1}{2}\log m + \frac{1}{2}\log 2\pi + \frac{1}{12})\, (O^+(k_C + k_X + k_Y) - O^+(k+2)) \\
& + mH(C_M, X_M, Y_M) - mH(C_M) - mH(X_M) - mH(Y_M)
\end{aligned}$$

Let us now relax the approximations in order to simplify the expression. Let use notice that $k_C + k_X + k_Y \leq k + 2$. We finally use the following bound $(1/2 \log 2\pi + 1/12) \approx 1.002 < 2$, and get

$$\begin{aligned}
c(M) - c(M_\emptyset) = {} & O^+(n \log k_C) - O^+(k \log k) + O^+(n \log n) \\
& + (k-1)\log(m+k) - O^+(n \log(m+n)) \\
& + (\frac{1}{2}\log m + 2)(O^+(k_C + k_X + k_Y) - O^+(k)) + O^\pm(n) \\
& + m(H(C_M, X_M, Y_M) - H(C_M) - H(X_M) - H(Y_M)).
\end{aligned}$$

Finally, we get the following approximation of $c(M) - c(M_\emptyset)$.

$$|(c(M) - c(M_\emptyset)) - a(k, m)| < b(k_C, k_X, k_Y, k, n, m)$$

where

$$\begin{aligned} a(k,m) =& m(H(C_M, X_M, Y_M) - H(C_M) - H(X_M) - H(Y_M)) \\ &+ (k-1)\log(m+k) \end{aligned}$$

and

$$\begin{aligned} b(k_C, k_X, k_Y, k, n, m) =& n(1 + |\log \frac{1}{k_C}(1 + \frac{m}{n})|) \\ &+ (\frac{1}{2}\log m + 2)(k+2) + k\log k. \end{aligned}$$

□

References

[1] D. Bosq, Linear Processes in Function Spaces: Theory and Applications (Lecture Notes in Statistics), Springer, 2000.
[2] J. Ramsay, B. Silverman, Applied Functional Data Analysis: Methods and Case Studies, Springer-Verlag Inc, 2002.
[3] J. Ramsay, B. Silverman, Functional Data Analysis, Springer Series in Statistics, Springer, 2005.
[4] T. Tarpey, K. Kinateder, Clustering functional data, Journal of Classification 20 (2003) 093–114.
[5] J. Peng, H. Müller, Distance-based clustering of sparsely observed stochastic processes, with applications to online auctions, The Annals of Applied Statistics 2 (3) (2008) 1056–1077.
[6] L. Sangalli, P. Secchi, S. Vantini, V. Vitelli, K-mean alignment for curve clustering, Computational Statistics & Data Analysis 54 (5) (2010) 1219–1233.
[7] R. Agrawal, C. Faloutsos, A. Swami, Efficient similarity search in sequence databases, in: D. Lomet (Ed.), Proceedings of the 4th international conference of foundations of data organization and algorithms (FODO), Springer Verlag, Chicago, Illinois, 1993, pp. 69–84.
[8] K. Chan, A. Fu, Efficient time series matching by wavelets, in: ICDE, 1999, pp. 126–133.
[9] F. Chamroukhi, A. Samé, G. Govaert, P. Aknin, A hidden process regression model for functional data description. application to curve discrimination, Neurocomputing 73 (7-9) (2010) 1210–1221.
[10] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning, Springer, 2001.
[11] C. Deboor, A practical guide to splines, Springer, 2001.
[12] C. Abraham, P. Cornillon, E. Matzner-Lober, N. Molinari, Unsupervised curve clustering using b-splines, Scandinavian journal of statistics 30 (3) (2003) 581–595.
[13] P. Smyth, Clustering sequences with hidden markov models, in: M. Mozer, M. Jordan, T. Petsche (Eds.), Advances in neural information processing systems, Vol. 9, The MIT Press, 1997, pp. 648–654.
[14] P. Smyth, Probabilistic model-based clustering of multivariate and sequential data, Proceedings of artificial intelligence and statistics (1999) 299–304.
[15] F. Rossi, B. Conan-Guez, A. E. Golli, Clustering functional data with the SOM algorithm, in: Proceedings of the ESANN, 2004, pp. 305–312.
[16] G. Hébrail, B. Hugueney, Y. Lechevallier, F. Rossi, Exploratory Analysis of Functional Data via Clustering and Optimal Segmentation, Neurocomputing / EEG Neurocomputing 73 (7-9) (2010) 1125–1141.
[17] F. Ferraty, P. Vieu, Nonparametric Functional Data Analysis: Theory and Practice, Springer Verlag, 2006.
[18] C. Crambes, L. Delsol, A. Laksaci, Robust nonparametric estimation for functional data, Journal of Nonparametric Statistics 20 (7) (2008) 573–598.
[19] T. Gasser, P. Hall, B. Presnell, Nonparametric estimation of the mode of a distribution of random curves, Journal of the Royal Statistical Society 60 (1998) 681–691.
[20] G. Delaigle, P. Hall, Defining probability density for a distribution of random functions, Annals of Statistics 38 (2) (2010) 1171–1193.
[21] T. W. Liao, Clustering of time series data-a survey, Pattern Recognition 38 (2005) 1857–1874.
[22] M. Boullé, Data grid models for preparation and modeling in supervised learning, Microtome Publishing, 2011, pp. 99–130.
[23] A. Tikhonov, V. Arsenin, Solution of Ill-posed Problems, John Wiley & Sons, 1977.
[24] J. Ramsay, Kernel smoothing approaches to nonparametric item characteristic curve estimation, Psychometrika 56 (4) (1991) 611–630.
[25] L. Devroye, L. Györfi, G. Lugosi, A probabilistic theory of pattern recognition, Springer, 1996.
[26] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, Knowledge discovery and data mining: towards a unifying framework, in: KDD, 1996, pp. 82–88.
[27] M. Boullé, Multivariate data grid models for supervised and unsupervised learning, Tech. Rep. NSM/R&D/TECH/EASY/TSI/5/MB, France Telecom R&D, http://perso.rd.francetelecom.fr/boulle/publications/BoulleNTTSI5MB08.pdf (2008).
[28] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, R. Wirth, CRISP-DM 1.0 : step-by-step data mining guide (2000).
[29] M. Boullé, Recherche d'une représentation des données efficace pour la fouille des grandes bases de données, Ph.D. thesis, Ecole Nationale Supérieure des Télécommunications, http://perso.rd.francetelecom.fr/boulle/publications/BoulleThesis07.pdf (2007).
[30] M. Abramowitz, I. Stegun, Handbook of mathematical functions, Dover Publications Inc., New York, 1970.
[31] C. Shannon, A mathematical theory of communication, Tech. Rep. 27, Bell systems technical journal (1948).
[32] J. Rissanen, Modeling by shortest data description, Automatica 14 (1978) 465–471.
[33] T. Cover, J. Thomas, Elements of information theory, Wiley-Interscience, New York, NY, USA, 1991.
[34] M. Boullé, Bivariate data grid models for supervised learning, Tech. Rep. NSM/R&D/TECH/EASY/TSI/4/MB, France Telecom R&D, http://perso.rd.francetelecom.fr/boulle/publications/BoulleNTTSI4MB08.pdf (2008).
[35] P. Hansen, N. Mladenovic, Variable neighborhood search: principles and applications, European Journal of Operational Research 130 (2001) 449–467.

[36] M. Boullé, A parameter-free method for clustering functional data, Tech. rep., France Telecom R&D, No FT/RD/TECH/11/01/76 (2011).
[37] F. Frappart, S. Calmant, M. Cauhopé, F. Seyler, A. Cazenave, Preliminary results of envisat ra-2-derived water levels validation over the amazon basin, Remote Sensing of Environment 100 (2) (2006) 252–264.
[38] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Proceedings of the IEEE, Vol. 86, 1998, pp. 2278–2324.
[39] D. Erhan, A. Courville, Y. Bengio, P. Vincent, S. Bengio, Why does unsupervised pre-training help deep learning?, Journal of Machine Learning Research 11 (2010) 625–660.
[40] O. Chapelle, B. Schölkopf, A. Zien, Semi-Supervised Learning, MIT Press, Cambridge, MA, 2006.