A Parameter-Free Method for Clustering Functional Data

Note technique

Référence : **Autre référence :** **Version :** **Date d'édition :**	**FT/RD/TECH/11/01/76** 1.0 11 janvier 2011	*Vérifié par : Fabrice Clérot* *Affiliation : TECH/ASAP* Le : 11 janvier 2011
Auteurs :	Boullé Marc TECH/ASAP	*Approuvé par : Patrice Soyer* *Affiliation : TECH/ASAP* Le : 11 janvier 2011

Résumé :

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 forms a nonparametric estimator of the joined 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 consistency of the approach is assessed using controlled experiments with artificial datasets, and its practical interest for functional data exploratory analysis is presented on three real world datasets.

Mots clés : Functional data, Exploratory analysis, Clustering, Bayesianism, Model Selection, Density estimation

Thème : 0100 - Théories de l'information, des communications et du signal

A Parameter-Free Method for Clustering Functional Data

Marc Boullé MARC.BOULLE@ORANGE-FTGROUP.COM
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 forms a nonparametric estimator of the joined 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 consistency of the approach is assessed using controlled experiments with artificial datasets, and its practical interest for functional data exploratory analysis is presented on three real world datasets.

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

1. Introduction

Functional data analysis (Bosq, 2000; Ramsay and Silverman, 2002, 2005) 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 problem with functional data is that of data representation, with a preprocessing task of representing the curves by of 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. 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 (Agrawal et al., 1993), discrete wavelet transform (Chan and Fu, 1999) or spline basis functions (Hastie et al., 2001; Deboor, 2001). In (Smyth, 1997, 1999), 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 (Rossi et al., 2004), the self-organizing map (SOM) clustering algorithm is applied to functional data equipped with a similarity matrix. In (Hébrail et al., 2010), 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 requires both fixing some function parameters, such 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 (Ferraty and Vieu, 2006; Crambes et al., 2008), 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 (Gasser et al., 1998; Delaigle and Hall, 2010), 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.

In this paper, we propose a novel exploratory method for functional data, based on data grid models (Boullé, 2010). 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 data grid of cells, which is a nonparametric estimator of the joined density of all the variables. 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 joined density estimation data grid models. 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 joined 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 (Hébrail et al., 2010), smoothness as in regularization (Tikhonov and Arsenin, 1977; Ramsay, 1991) or capacity as in learning theory (Devroye et al., 1996). 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 approach for data grid models and apply it to joint density estimation and clustering for functional data. We illustrate the approach and assess its performance using artificial data in Section 3. We present experimental results on three real world datasets in Section 4, and show what kind of exploratory analysis can be performed. We show in Section 5 how to a "natural" distance between clusters of curves emerges from the approach and provides the

basis for insightful post-processing of the clustering results. Finally, we give a summary in Section 6.

2. MODL Approach for Functional Data Clustering

In this section, we first summarize the principles of data grid models introduced in (Boullé, 2010) 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" (Fayyad et al., 1996). 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 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 (Boullé, 2008b, 2010) have been introduced for the data preparation phase of the data mining process (Chapman et al., 2000), 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 $\mathcal{C}$ 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 consist of $n = 2$ curves with respectively $m_1 = 4$ and $m_2 = 5$ points:

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

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

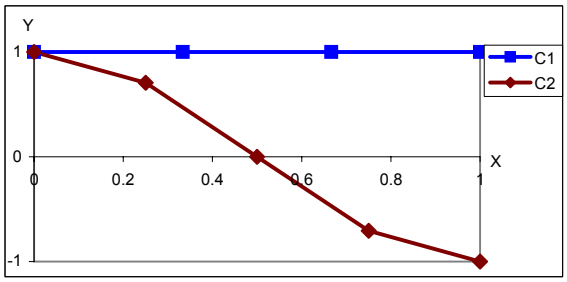

Figure 1: Two sample curves.

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.

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.

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 peacewise 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 3.1.

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 peacewise constant joint density estimation of the curves and points. The finest data grid consists of one curve per cluster and one point value per interval, whereas the coarsest one contains all the points of all the curves in one single cell. 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.

- $\mathcal{C}$: collection of curves
- $\mathcal{P}$: point dataset containing all points of $\mathcal{C}$ in tabular format
- C: curve variable
- X, Y: variables for the point dimensions
- $n = |\mathcal{C}|$: 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
- $k_C(i)$: 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

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

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

$$m_{i_C} = \sum_{j_X=1}^{k_X} \sum_{j_Y=1}^{k_Y} m_{i_C j_X j_Y},$$
$$m_{i_X} = \sum_{i_C=1}^{k_C} \sum_{j_Y=1}^{k_Y} m_{i_C j_X j_Y},$$
$$m_{i_Y} = \sum_{i_C=1}^{k_C} \sum_{j_X=1}^{k_X} m_{i_C j_X j_Y}.$$

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).

In order to select the best model, we apply a Bayesian approach, using the prior distribution on the model parameters described in Definition 2.

Definition 2 *The prior for the parameters of functional data clustering model are chosen hierarchically and uniformly at each level:*

- *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,*

- *for a given number k_C of clusters, every partition of the n curves into k_C clusters are equiprobable,*
- *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,*
- *for a given cluster of curves, every distribution of the points in the cluster on the curves of the cluster are equiprobable,*
- *for a given interval of X (resp. Y), every distribution of the ranks of the X (resp. Y) values of points are equiprobable.*

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 (Boullé, 2007).

Theorem 3 *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*

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

$B(n, k)$ 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 (see Abramowitz and Stegun, 1970), which stands for the number of ways of partitioning a set of n elements into i nonempty subsets.

The first line in Formula 1 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 Relation with Information Theory

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 1, the cost $c(M_\emptyset)$ of the null model (its value according to evaluation criterion 1)

reduces to

$$c(M_\emptyset) = \log n + 2\log m + \log \binom{m+n-1}{n-1} + \log \frac{m!}{m_1!m_2!\ldots m_n!} + 2\log m! \tag{2}$$

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.

Entropy of the null model. To get an asymptotic evaluation of the cost of the null model, we now introduce the Shannon entropy $H(V)$ (Shannon, 1948) of a discrete variable V, $H(V) = -\sum_{v \in V} p(v) \log p(v)$. Let us consider points as statistical instances, described by the curve variable C having n values and the dimension variables X and Y having m ranks. As m_i stands for the number of points of curve i, the probability that a point belongs to curve i can be estimated by $\frac{m_i}{m}$. For the dimension variables X and Y, the probability of each rank of a value among m potential ranks is $\frac{1}{m}$. We thus get $H(C) = -\sum_{i=1}^{n} \frac{m_i}{m} \log \frac{m_i}{m}$ and $H(X) = H(Y) = -\sum_{j=1}^{m} \frac{1}{m} \log \frac{1}{m}$.

Using the approximation $\log n! = n(\log n - 1) + O(\log n)$ based on Stirling's formula, the cost of the null model is asymptotically equivalent to m times the Shannon entropy of the curve variable C and the dimension variables X and Y:

$$c(M_\emptyset) = m(H(C) + H(X) + H(Y)) + O(\log m). \tag{3}$$

Coding length of the null model. As negative log of probabilities can be interpreted as a coding length (Shannon, 1948), criterion 2 can be interpreted as the coding length of dataset $\mathcal{P}$. The first line of criterion 2 encodes the number of curves and the number of intervals per dimension, then the parameters of the multinomial distribution of the m points on the n curves. The second line encodes the actual curve related to each point, owing to the negative log of a multinomial term, followed by the actual ranking of the values per point dimension.

Coding length of functional data clustering models. Extending the coding length interpretation to any model, our model selection technique is closely related to the minimum description length (MDL) approach (Rissanen, 1978; Hansen and Yu, 2001; Grünwald et al., 2005), which aims at approximating the Kolmogorov complexity (Li and Vitanyi, 1997) for the coding length of the point dataset $\mathcal{P}$ (which is equivalent to that of the collection of curves $\mathcal{C}$). The Kolmogorov complexity is the length of the shortest computer program that encodes the data. The prior terms in Formula 1 represent the coding length of the functional data clustering model parameters whereas the likelihood terms represent the coding length of the data (the points) given the model.

Robust joint density estimation in collections of curves. Overall, our prior approximates the Kolmogorov complexity of the functional data clustering model and our conditional likelihood encodes the points given the model. In our approach, the choice of

the null model corresponds to the lack of reliable structure in the collection of curves. The coding length of the null model is asymptotically equivalent to the Shannon entropy of the distribution of the curves and of the ranks of the dimension variables (cf. Formula 3), which corresponds to a basic encoding of the points, without any use of structure in the collection of curves. This is close to the idea of Kolmogorov, who considers data to be random if its algorithmic complexity is high, that is if it cannot be compressed significantly. This makes our approach very robust, since detecting reliable structures using functional data clustering models is necessarily related to a coding length better than that of the null model, thus to non random patterns according Kolmogorov's definition of randomness. This robustness has been confirmed using extensive experiments in the case of univariate data preparation for supervised data mining (Boullé, 2006, 2005) and is evaluated in the case of functional data in Section 3.

2.5 Optimization Algorithm

Functional data clustering models are no other than data grid models (Boullé, 2007) 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 1 is optimal, the design of sophisticated optimization algorithms is both necessary and meaningful. Such algorithms are described in (Boullé, 2007, 2008a). 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 $\mathrm{O}(m)$ memory complexity and $\mathrm{O}(m\sqrt{m}\log m)$ time complexity.

In this section, we give an overview of the optimization algorithms which are fully detailed in (Boullé, 2007), 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 1. 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 adjacent clusters or intervals, for the curve and 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.

Each evaluation of the criterion for a data grid model requires $\mathrm{O}(nm^2)$ time, since the initial model contains up to nm^2 cells (see Formula (1)) in the case of the maximal model M_{Max}. Each step of the algorithm relies on $\mathrm{O}(n^2)$ evaluations of merges of clusters of curves and $\mathrm{O}(2m)$ evaluation of merges of intervals, and there are at most $\mathrm{O}(n+2m)$ 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 $\mathrm{O}(n^4m^2+n^3m^3+nm^4)$ using a straightforward implementation of the algorithm. However, the method can be optimized in $\mathrm{O}(m\sqrt{m}\log m)$ time, as demonstrated in (Boullé, 2007). The optimized algorithm mainly

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**

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 1, 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 pre-processing and post-processing 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 time complexity of $O(m\sqrt{m}\log m)$ 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 (Hansen and Mladenovic, 2001), which mainly benefits from multiple runs of the algorithms with different random initial solutions. 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.

The optimization algorithms summarized above have been extensively evaluated in (Boullé, 2008a), 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. Evaluation on Artificial Data

We first illustrate the behavior of our approach using a toy dataset, then evaluate our method using a complex artificial dataset, where the true clustering consist of hundreds of curves belonging to tens of noisy clusters, some of them hard to discriminate.

3.1 Illustration with Two Simple Curves

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.

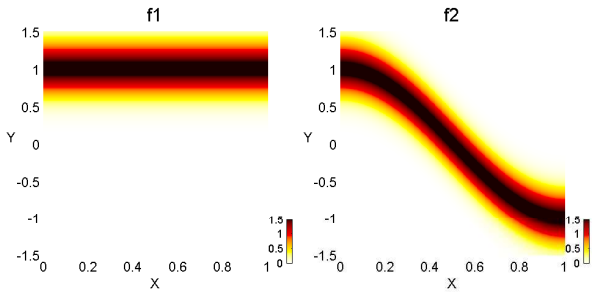

Figure 2: Two sample functions f_1 and f_2 drawn with their conditional density $d(y|x)$.

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 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 ten curves related to the $f1$ and $f2$ 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 and 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.

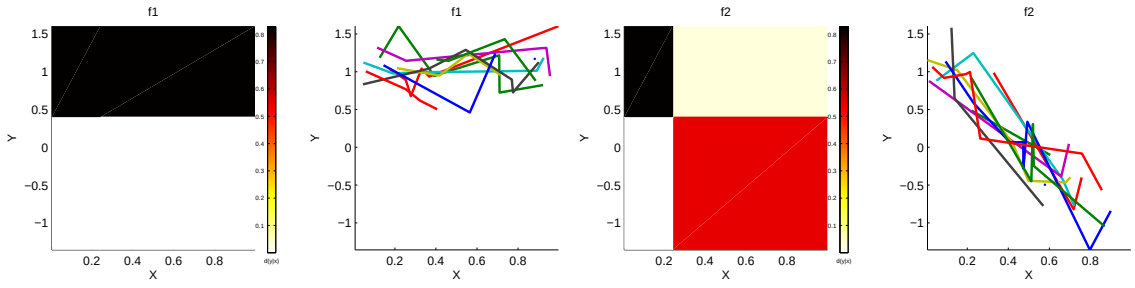

Figure 3: Estimation of the conditional density $d(y|x)$ and discovered clusters of curves, with 5 points per curve.

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.

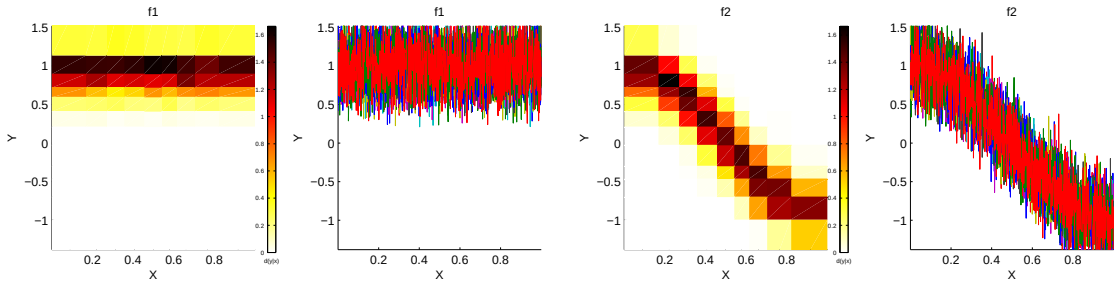

Figure 4: Estimation of the conditional density $d(y|x)$ and discovered clusters of curves, with 1000 points per curve.

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. 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 produce spurious clusters whatever the sample size.

3.2 Evaluation on a Complex Curve Artificial Dataset

The purpose of this controlled experiment is to study the ability of our method to discriminate complex clusters and to avoid the detection of spurious patterns for datasets of increasing sizes. We introduce a space of curves defined on the domain of x values $[0, 1]$, with two shape parameters a and b and an additive white Gaussian noise $N(0, \sigma)$:

$$C(a, b, \sigma) : y = \sin a\pi x + \cos b\pi x + N(0, \sigma). \tag{4}$$

The two functions in Section 3.1 correspond to $f1 : C(0, 0, 0.25)$ and $f2 : C(0, 1, 0.25)$. We consider 48 families of curves, using $a \in \{0, 1, 2, 3\}$, $b \in \{0, 1, 2, 3\}$ and $\sigma \in \{0.25, 0.5, 1.0\}$. The conditional density $d(y|x)$ of each family is drawn in Figure 5.

For each family, we generate 10 curves, for a total a 480 curves, representing 16 shapes with three levels of noise. We generate a dataset $\mathcal{P}$ of one million points, on average 2000 per curve. Each point is a triple of values with a randomly chosen curve (among 480), a random x value (on domain $[0, 1]$) and a y value generated according to the function related to the curve. Let us notice that this dataset is difficult to analyze for any functional data analysis method based on parametric or semi-parametric regression of each curve for the following reasons:

- the considered curves are very noisy and therefore do not meet the assumption that they are intrinsically smooth, like in (Ramsay and Silverman, 2005),

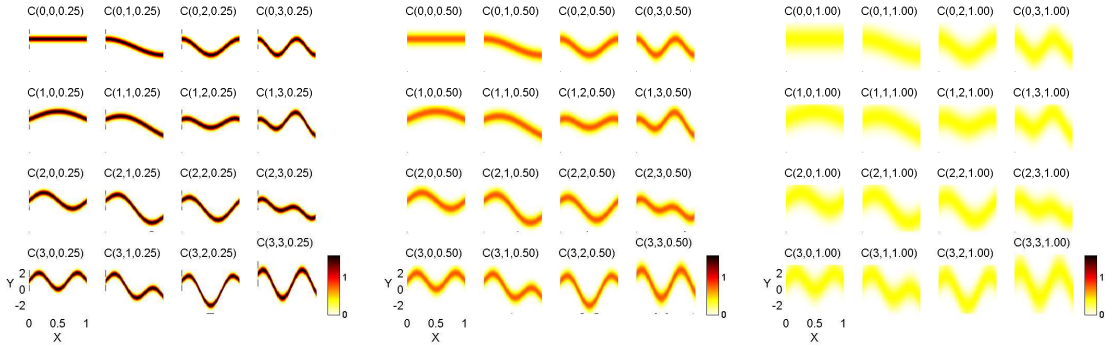

Figure 5: Artificial curve patterns drawn according their conditional density $d(y|x)$. 16 curve shapes with standard deviation 0.25, 0.5 and 1.0

- regression methods estimate the expectation of the function and are prone to overfitting in case a noisy data with problems to discriminate different levels of noise for the same shapes,
- most functional data analysis approaches require parameter tuning, which is not suitable to study the behavior of a method w.r.t datasets of varying sizes.

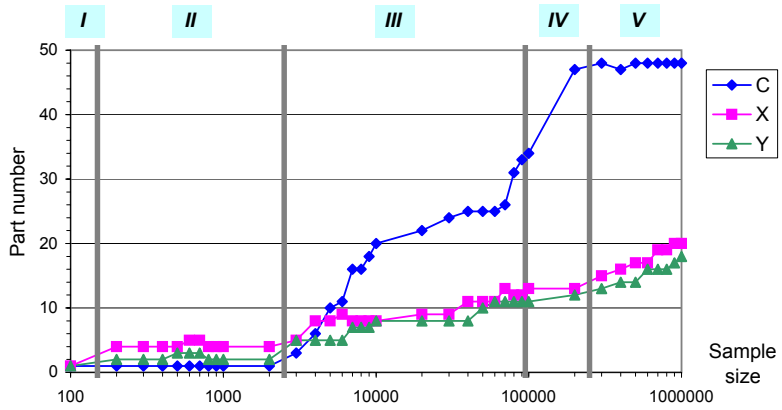

Figure 6: Artificial curve patterns: number of clusters and of intervals of X, Y of the data grids obtained from increasing size of the point dataset.

Like in Section 3.1, we apply our method on subsets of $\mathcal{P}$ with increasing size. Figure 6 reports the numbers of clusters of curves and the numbers of intervals per dimension X, Y for subsets of size 100 to 1 000 000. Interestingly, five phases can be distinguished to qualify the behavior of the method.

1. In phase I, up to 100 points, there is not enough data to discriminate any pattern and the method builds one single cluster of curves with one single interval per dimension, like in the independence case (see Sections 2.4 and 3.1).

2. In phase II, from 200 to 2000 points, the method is able to detect the global shape of the whole dataset, but not to discriminate any cluster of curves. It groups all the curves in one single cluster, but discretizes the dimensions with four interval for X and two for Y. Figure 7 displays on the left the estimated conditional density $d(y|x)$

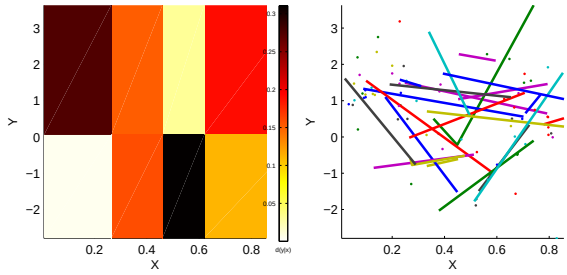

Figure 7: Point dataset of size 200: one single cluster.

and on the right the curves, most of them sampled with one single point. The overall shape of all curves in the dataset is to start with positive values of y for small values of x, then decrease down to a minimum with negative values of y around $x = 0.5$, with an almost uniform distribution of y for x before and after the minimum.

3. In phase III, from 3000 to 100 000 points, the method benefits from the growing number of points to build a more and more precise summary of the dataset with an increasingly detailed and precise partition of the 480 curves into clusters and an improving joint discretization of the dimensions X and Y. Figure 8 shows the three

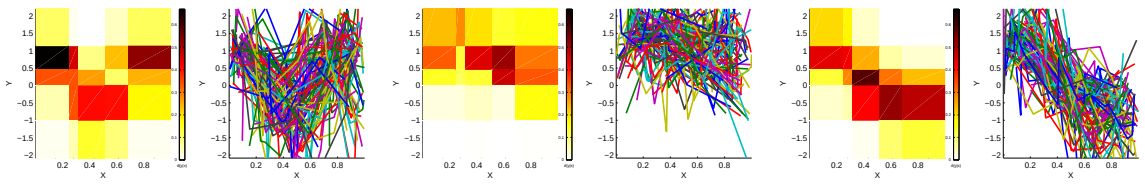

Figure 8: Point dataset of size 3000 (6 points on average per curve): three clusters.

clusters detected by our method with 3000 points, on average 6 points per curve. The dimensions X, Y are discretized using a $5 * 5$ grid, which provides an insightful summary of the three clusters. The first cluster groups strongly concave curves, the second one slightly convex curves with mostly positive y values, and the last one sharply decreasing curves. With so few points per curve, the partition of the curves is not fully consistent with the true curves families: several curves belonging to the same family fall into different clusters. With datasets of growing size, the number of

clusters increases and their purity w.r.t. the true curves families gets better. With a dataset of 100 000 points, the 48 true curve families are partitioned into 34 clusters in a consistent way, owing to a joint discretization of the dimensions X, Y into a $13 * 11$ grid. Twenty clusters consist of the 10 curves of one single curve family, and the other fourteen clusters contain the 20 curves of two close families. Figure 9 displays the 20

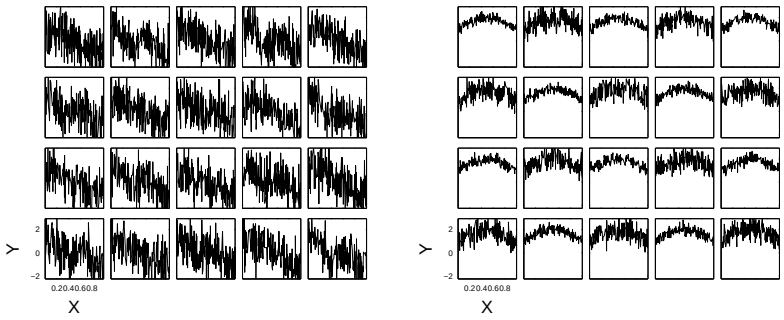

Figure 9: Point dataset of size 100 000: two sample clusters of 20 curves. On the left, mixture of two close shapes with high variance ($C(0, 1, 1.0)$ and $C(2, 3, 1.0)$), on the right, mixture of the same shape with small and medium variance ($C(1, 0, 0.25)$ and $C(1, 0, 0.5)$).

curves of two of the mixture clusters, one with similar shapes but high variance, the other one with the same shape but either small or medium variance.

4. In phase IV, from 100 000 to 300 000 points, the clusters are always consistent w.r.t. the curves families: they consist of all the curves of either one or two curve families. The discrimination of the true curve families gets better and better, until a fine grain grid of $15 * 13$ is exploited for 300 000 points, that enables the perfect discrimination of the 48 true curves families. Figure 10 displays the two clusters the most difficult

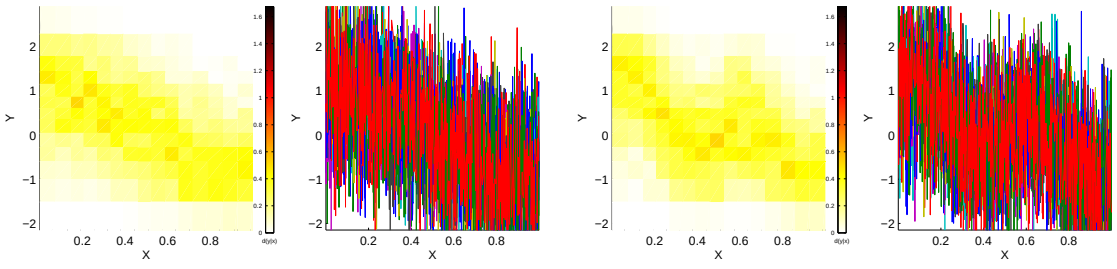

Figure 10: Point dataset of size 300 000: curves families $C(0, 1, 1.0)$ and $C(2, 3, 1.0)$, the most difficult to discriminate.

to discriminate, with close curve shape and high variance.

5. Finally, in phase V, the method always discriminates the true curves families with an increased precision, up to a grid of $20 * 18$ for 1 000 000 points. Overall, the finest grain trivariate datagrid contains $48*20*18 = 17280$ cells, which mean that the 17280 multinomial distribution parameters $m_{i_{C}j_{X}j_{Y}}$ are estimated using the model selection approach presented in Section 2. Figure 11 displays the conditional density estimation

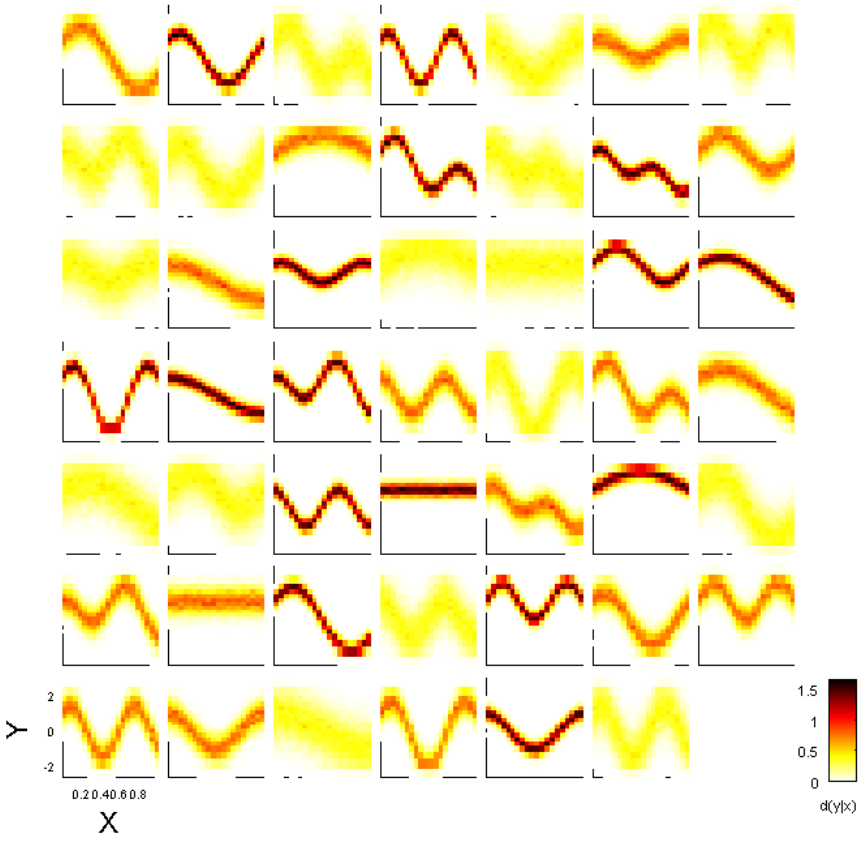

Figure 11: Point dataset of size 1 000 000: all the curve families are accurately discriminated.

for the 48 curves families, estimated for the whole dataset of 1 000 000 points.

Overall, the experiment demonstrates that the approach is both accurate and robust. It is able to approximate complex curve families provided that there is enough data, and never produces spurious clusters. The method is a nonparametric estimator of the joint probability of the curve and dimension variables. Unlike parametric methods where the parametric assumption may be questionable and the trade-off between coarse grain and fine grained summaries is a difficult task, the proposed approach automatically exploits the available data to build a summary of the dataset that is optimally accurate and robust.

4. Experimental Results on Real Data

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

4.1 Topex/Poseidon Satellite

The first dataset[1] detailed in (Frappart et al., 2006) 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 12 displays 10 curves chosen randomly from the dataset.

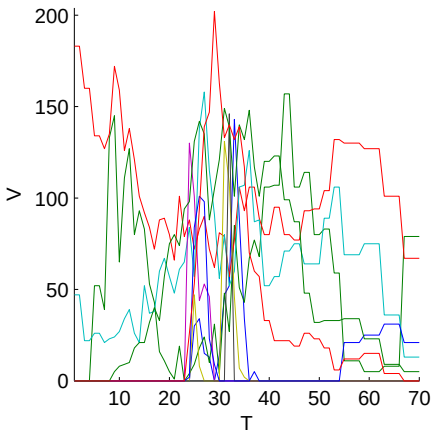

Figure 12: Topex/Poseidon dataset: 10 sample curves.

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 the conditional probability $p(v|t)$ on a bivariate discretization $7 * 4$ of the measure dimensions.

Figure 13 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.

Figure 14 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

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

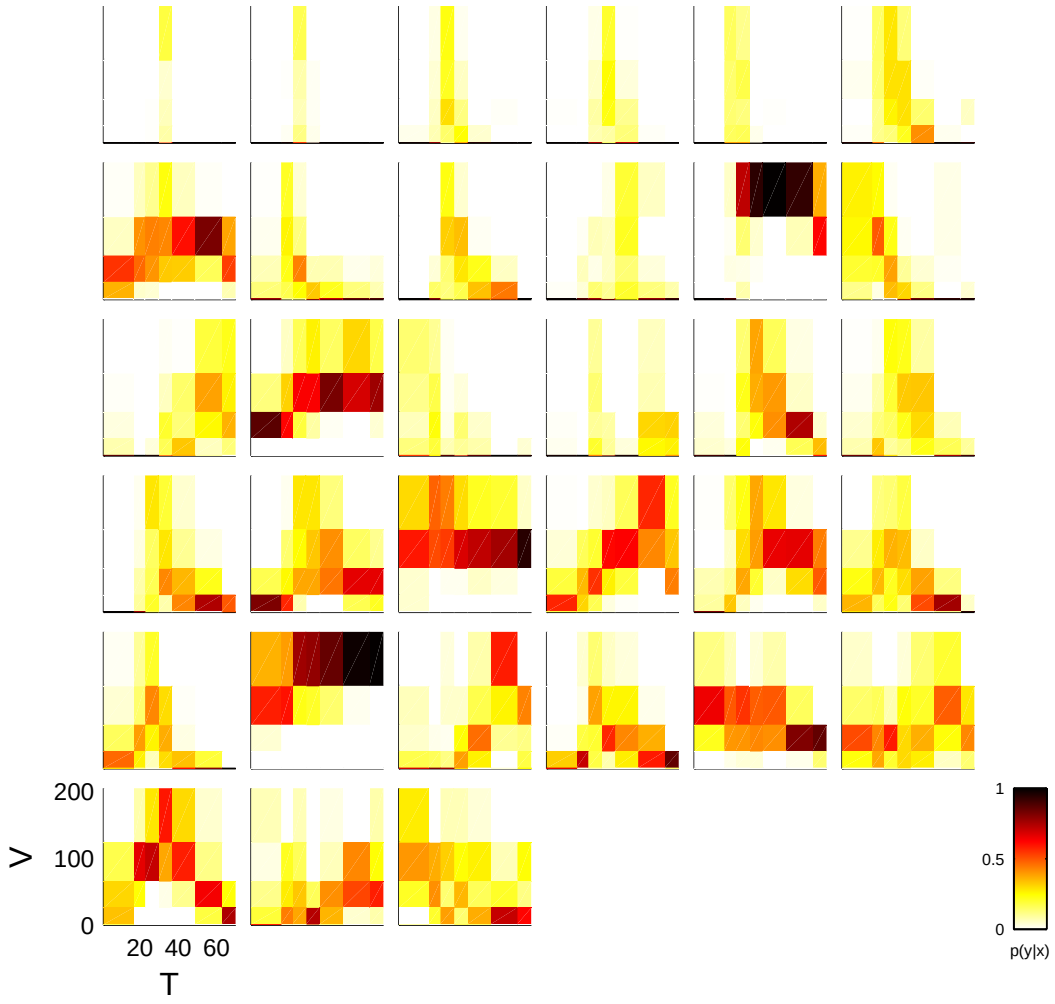

Figure 13: Topex/Poseidon dataset: all clusters.

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

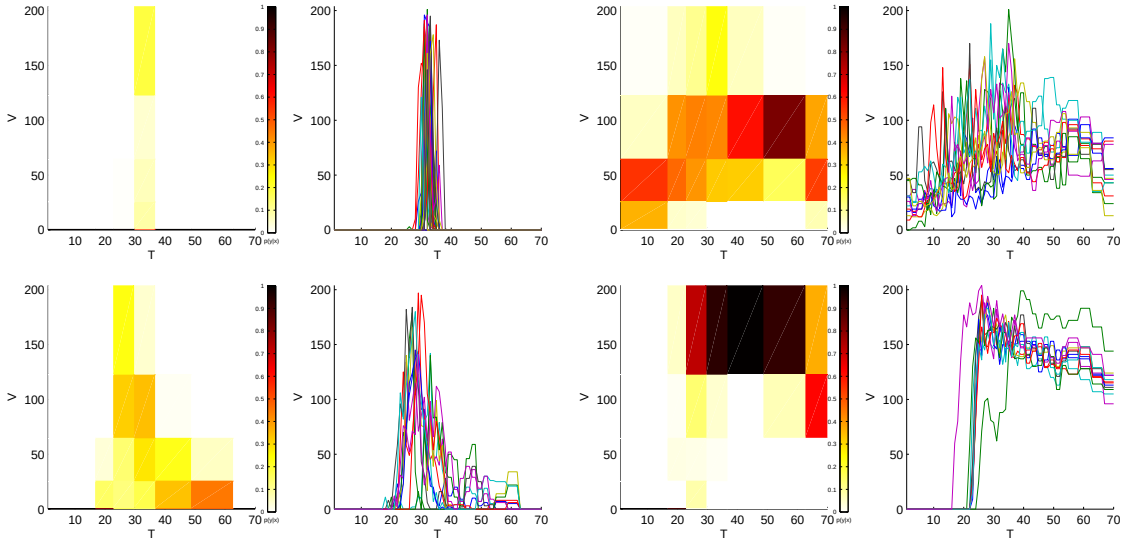

Figure 14: Topex/Poseidon dataset: four examples of clusters.

4.2 Electric Power Consumption

The second dataset[2] detailed in (Hébrail et al., 2010) consists in the electric power consumption recorded in a personal home during almost one year (349 days). Each curve consists in 144 measures which give the power consumption of one day at a 10 minutes sampling rate. Figure 12 displays 10 curves chosen randomly from the dataset.

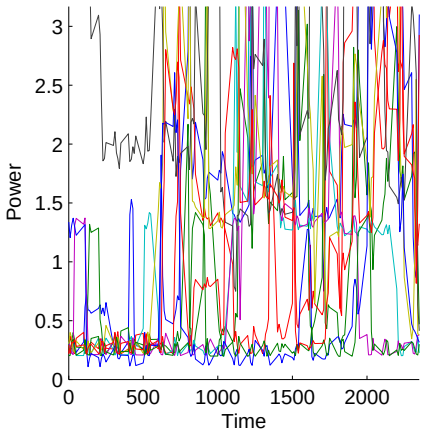

Figure 15: Power consumption dataset: 10 sample curves.

2. Dataset available at http://bilab.enst.fr/wakka.php?wiki=HomeLoadCurve

We reformat the original data as a three-dimensional dataset, and obtain a point dataset of $349 * 144 = 50256$ points with one curve variable, the recorded day, and two dimension variables, the time of the measure and the measured power. We apply the proposed method and obtain 60 clusters, summarized by the conditional probability $p(power|time)$ on a bivariate discretization with 7 intervals of time and ten intervals of power level.

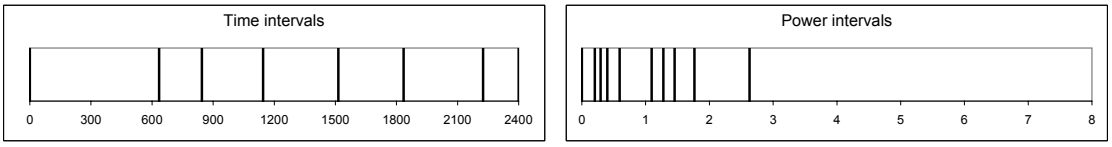

Figure 16: Power consumption dataset: discretization of the time and power variables.

The discretization of the dimension variables time and power, displayed in Figure 16, provides a first level of information. The time variable is discretized using seven intervals, which look consistent w.r.t. the usual periods of activities in a personal home: night, breakfast, morning, noon, afternoon, evening, and night again. The power variable is discretized using ten intervals, with a focus on the small levels of power consumption. In the following, we truncate the last wide interval in the displayed figures to focus on the discriminating patterns of power consumption. The fine grained discretization of the power variable makes sense to summarize complex behaviors resulting from many electrical appliances that might be switched on or off. It would be interesting to compare the bounds of the discretization intervals with the power of the electrical appliances available at the personal home related to this dataset.

Figure 17 displays six among the 60 clusters, with both the conditional probability $p(power|time)$ and all the curves related to each cluster. For example, the first cluster (top left figure) looks representative of a vacation period away from home: the power consumption is very low on average. It is noteworthy that the distribution is bimodal, with the power consumption mostly around 0.3 and with a small probability around 1.5. The curves of the cluster suggest that an electrical appliance is switched on for short periods of time, at random moments of the day. The second cluster (top right on Figure 17) looks typical of a week day, with low consumption during night and work hours, a sharp peak during breakfast and a wide peak in the evening. The third cluster (middle left) could be the signature of a week-end, with low consumption during the night and high consumption all day long, including the late evening period. The sixth cluster (bottom right) exhibits an unusual pattern, with high consumption during the night. The patterns identified by the method are discriminating and suggest that external data related to personal activity or usage of electrical appliances could be collected and correlated with the clusters to provide an adequate interpretation.

Figure 18 displays a summary of all the clusters. The number of combinations of low versus high power consumption for each of the seven time periods ($2^7 = 128$) provides a raw explanation of the high number of clusters (sixty) retrieved by the method. The theory behind the approach says that the number of clusters is optimal (within the effectiveness of the optimization algorithm), in the sense that more clusters would be less probable, leading to overfitting the data and creation of spurious clusters, and less clusters would result

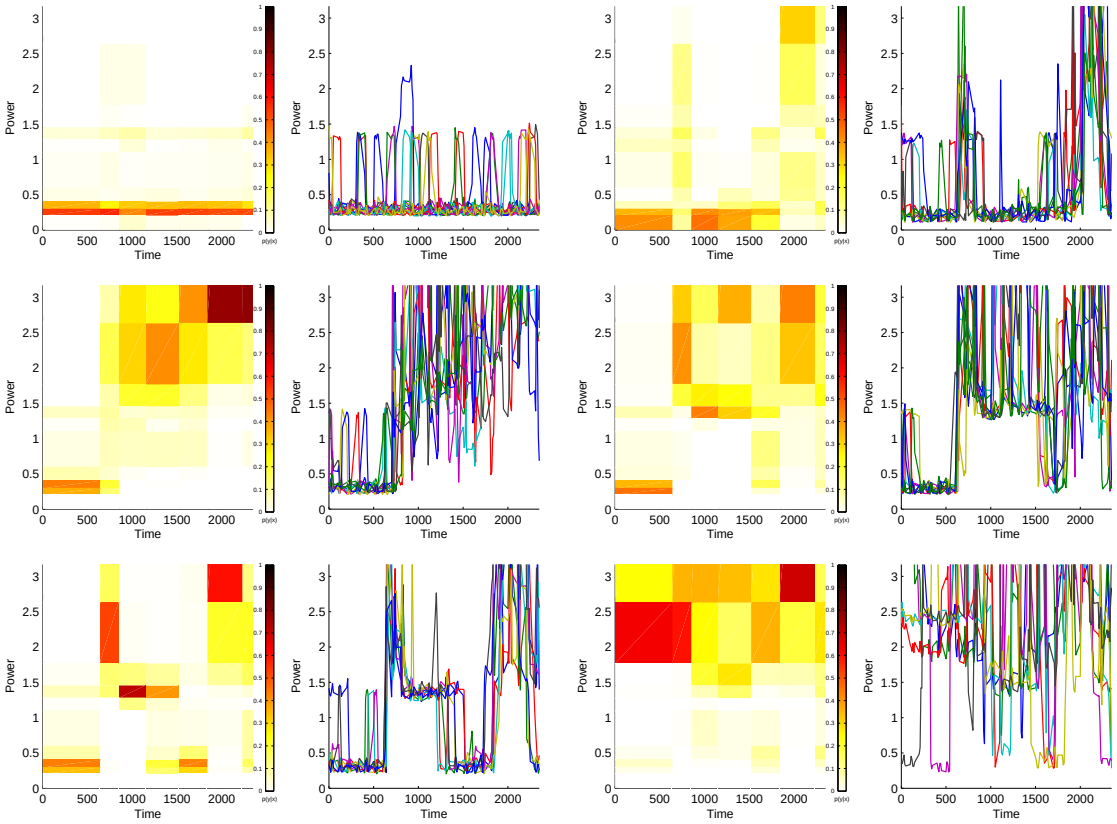

Figure 17: Power consumption dataset: six examples of clusters.

in a less probable explanation that underfit the data. It is noteworthy that the primary objective of the proposed approach is joint density estimation and that the partition of the curves into clusters is just a byproduct. In the case of a real world dataset, there is no reason of stopping the clustering process at a given grain level. We presume that with more data, for example with power curves sampled every minute instead of every 10 minutes, the method would potentially build one cluster per curve. Whereas this provides an accurate estimation of the joint probability of the dataset, this is no longer suitable for exploratory analysis and understandable explanation of the data. A simple solution to that issue is to set an upper-bound for the number of clusters and to constrain the optimization heuristics to retrieve the most probable clustering that fit this user parameter. This provides a trade-off between accuracy and understandability of the cluster. A possibly better solution is to let the method find the optimal clustering, then to post-process it using a hierarchical agglomerative algorithm. This is discussed in Section 5.

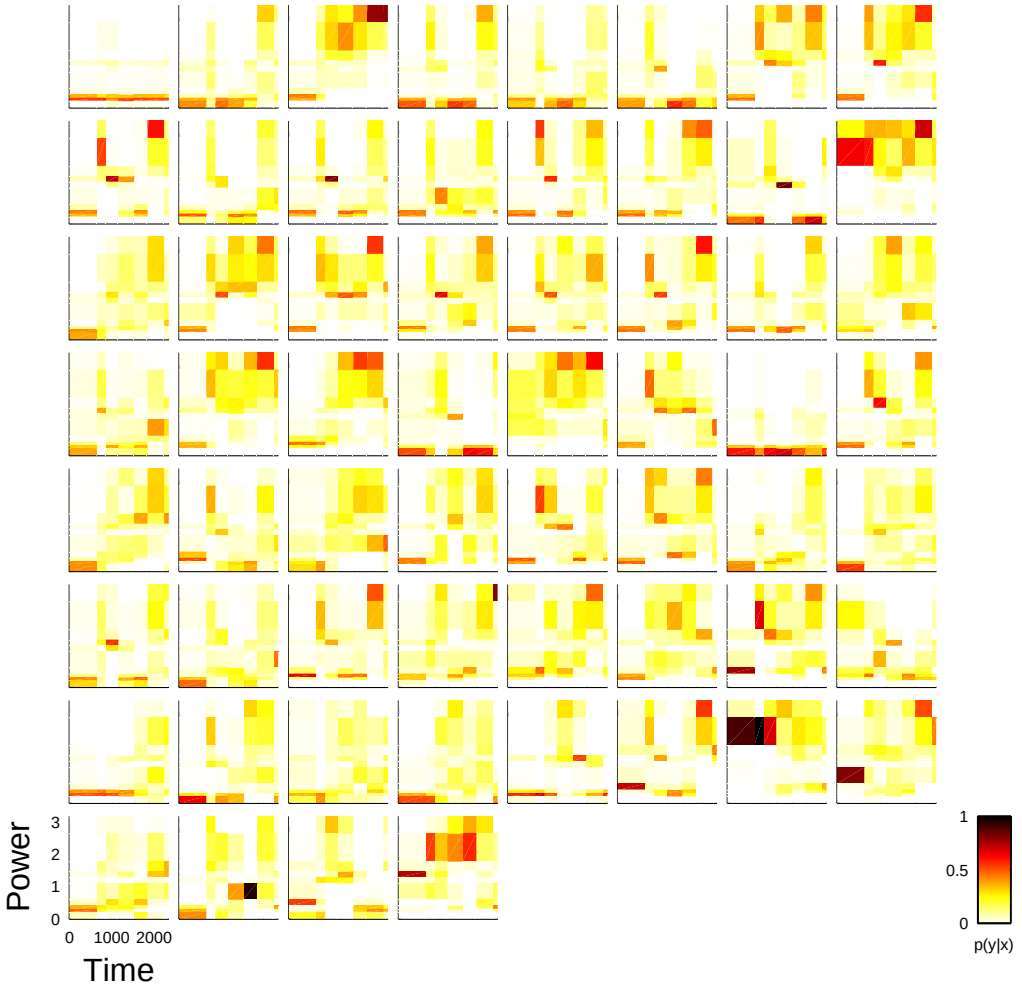

Figure 18: Power consumption dataset: all clusters.

4.3 MNIST Handwritten Digits

The third dataset[3] detailed in (Lecun et al., 1998) consists of 8-bit grayscale 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

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

space X, Y, leading to a point dataset $\mathcal{P}$ containing about 7.2 million points. Figure 12 displays 100 curves chosen randomly from this curve dataset.

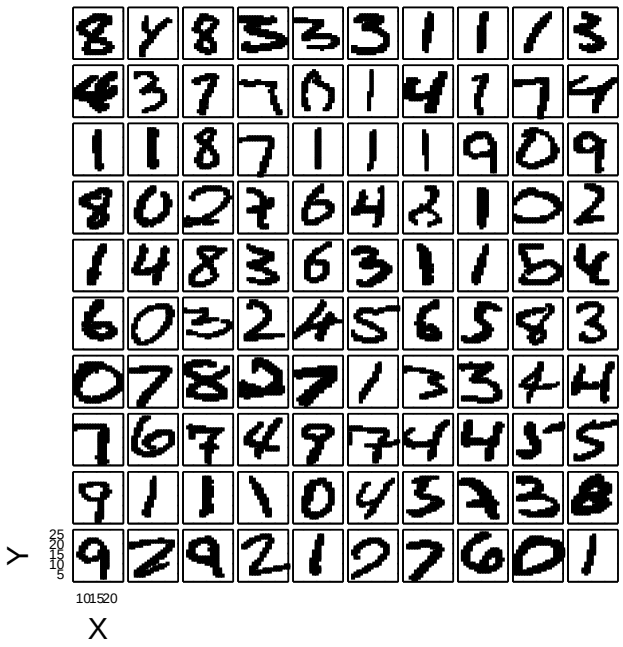

Figure 19: Handwritten digits datasets: 100 sample curves.

In this last 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 the task a exploratory analysis of functional data is multiple.

- This is a large dataset, two orders of magnitude above the Topex/Poseidon satellite and electrical power consumption datasets. 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, to our knowledge, it is the first time it is used for the task of exploratory analysis. 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.

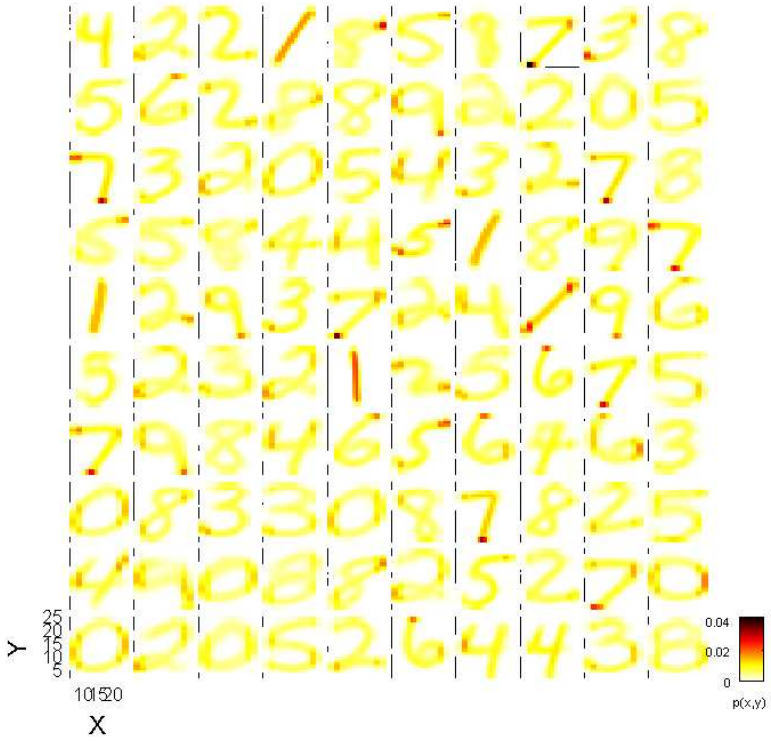

Figure 20: Handwritten digits datasets: 100 sample clusters.

We apply the method presented in Section 2 and 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 20 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 19.

Given this proximity of cluster summaries to digit shapes, we decide to reorganize the clusters according to their majority digit. For each cluster, we compute the number of curves per digit and assign the cluster to its majority digit. Figure 21 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,

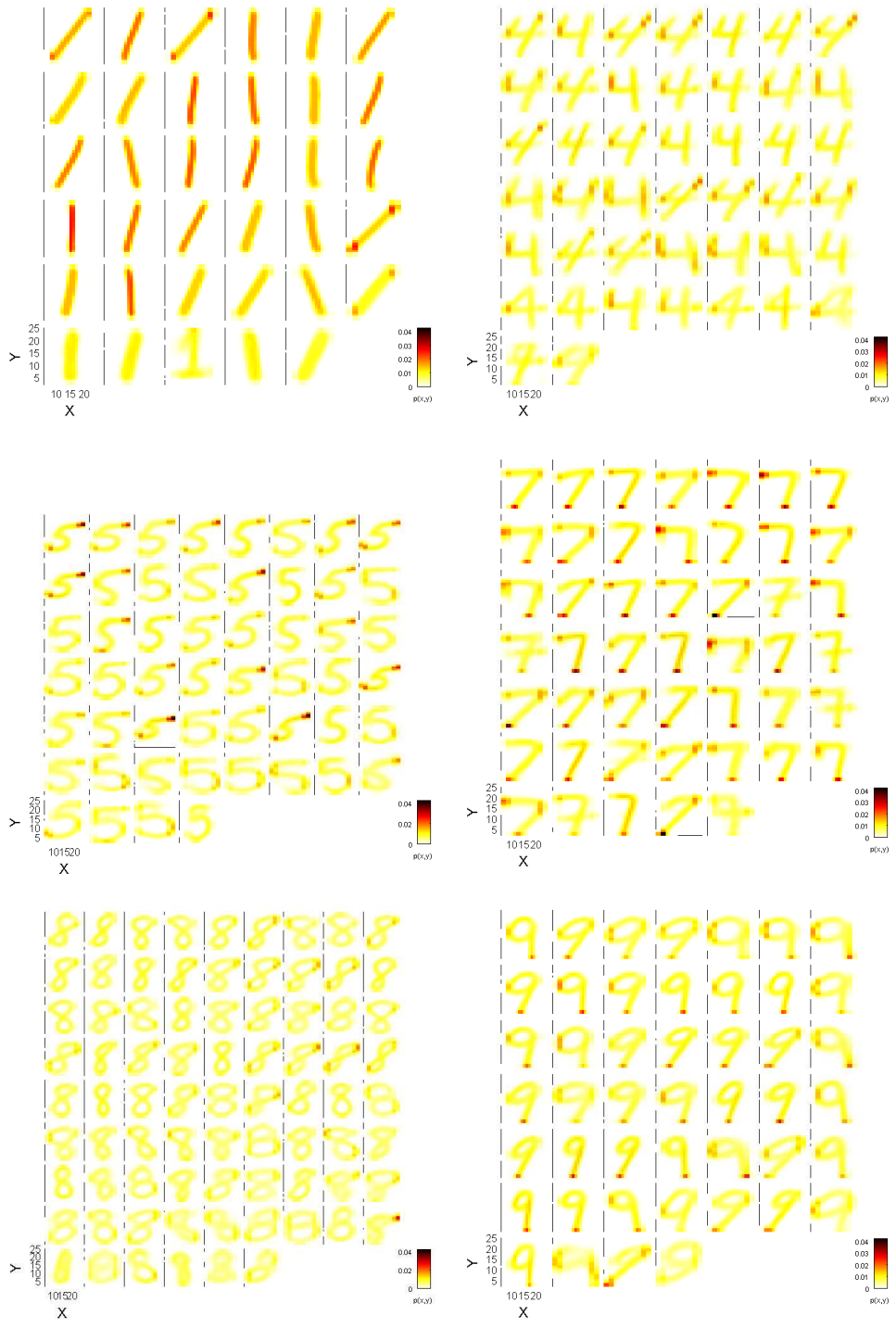

Figure 21: Handwritten digits datasets: clusters organized according to their majority digit, for digits “1”, “4”, “5”, “7”, “8”, “9”.

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 48 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 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.

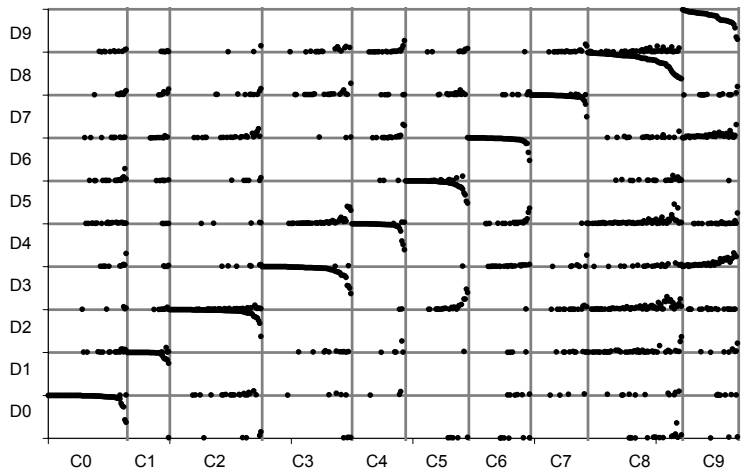

Figure 22: Handwritten digits datasets: distribution of digits per cluster.

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 22. 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" report 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 column "C8" report 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".

Figure 23 displays two clusters of "9". The first one contains a random subset of the 170 curves of the cluster, almost of them being labeled "9" (with two exceptions: one "4" and one "7"). The second one is one the most mixed among all the clusters, with 97 curves and only 35% of them being labeled "9" (precisely with one "1", three "2", twenty three "4", one "5", thirty "7", five "8" and thirty four "9").

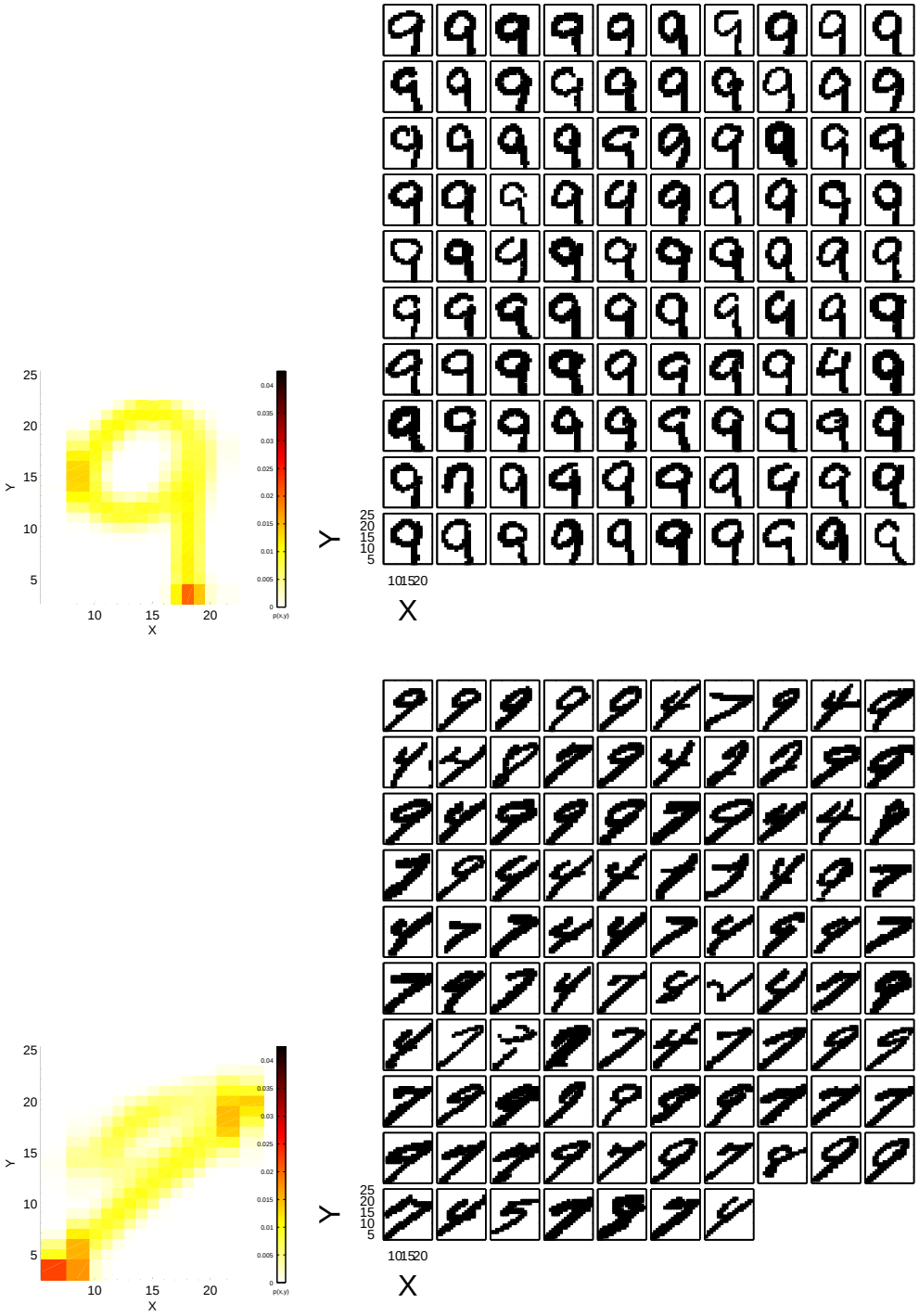

Figure 23: Handwritten digits datasets: two clusters related to digit "9", the one on the top is almost pure whereas the other one on the bottom is very noisy.

Finally, we evaluate the clusters as a preprocessing method for digit classification, using a semi-supervised learning setting (Chapelle et al., 2006). All the train and test examples are used without the class labels (the digits) to train the unsupervised clustering model and produce the clusters. Then, 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. To summarize, each image, initially described by a vector of $28 * 28 = 784$ pixels, is recoded with one single variable, the index of a cluster, and classified by counting inside each cluster. 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 3 neighbors is 5.0% without preprocessing and 2.4% with after deskewing. Two-layer neuron networks with 300 hidden units achieved an error rate of 4.7% without preprocessing and 1.6% 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. Finally, convolutional networks with up to seven layers and boosting obtained a 0.7% error rate. These first results were reported in a vast comparative benchmark (Lecun et al., 1998) 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.

One new issue arises in this context of using the clusters as a preprocessing for a classification task. In real world settings, semi-supervised learning cannot be applied and it is necessary to assign new unknown curves to the trained clusters. This will be discussed in Section 5.

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.

5. Introducing a Similarity between Clusters of Curves

This section is mainly intended to suggest and motivate future work. We show how a similarity can derived from our approach and bring new features that extend the possibilities of functional data exploratory analysis.

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

5.1 Similarity between Clusters of Curves

Most clustering techniques like K-means, Kohonen maps, or any agglomerative hierarchical algorithm are based on a similarity between instances. Determining such a similarity is a critical task, that results from many choices, such as the representation in the initial feature space, the use of a modeling technique to transpose the instances into a parameter space or the distance (Euclidian, Manhattan...).

We show here how the approach introduced in Section 2.3 can be used to derive a similarity between clusters of curves. The evaluation criterion $c(M)$ (see Formula 1) is related to the posterior probability of a functional data clustering model M. Let M_{Max} be the best model obtained using Algorithm 1, with k_C clusters of curves and a $j_X * j_Y$ bivariate discretization of the point dimensions and let $i_{C1}, i_{C2} \in \{1, \ldots, k_C\}$ be the indexes of two clusters. Let $M_{i_{C1} \cup i_{C2}}$ be the model resulting from the merge of the two clusters i_{C1} and i_{C2} of curves, keeping the same bivariate discretization of the point dimensions. We then suggest the following similarity between the clusters as:

$$\begin{aligned} \delta(i_{C1}, i_{C2}) =& c(M_{i_{C1} \cup i_{C2}}) - c(M_{Max}), \\ =& \log \frac{p(M_{Max}|\mathcal{P})}{p(M_{i_{C1} \cup i_{C2}})|\mathcal{P})}. \end{aligned} \tag{5}$$

Since the best model M_{Max} obtained using Algorithm 1 results from a bottom-up agglomerative heuristic, each merge can only decrease the criterion and we have $\delta(i_{C1}, i_{C2}) \geq 0$. Intuitively, if two clusters share a similar distribution on the X, Y space, the total coding length of the data (see criterion $c(M)$) is not much different between the cases where the clusters are coded jointly or separately, so that δ will be close to 0. On the opposite, mixing two clusters that have a significantly different distributions on X, Y results in an important loss of information and thus a large value of δ.

In the bottom-up agglomerative Algorithm 1, minimizing the cost $c(M)$ of a model by merging clusters of curves is exactly the same as merging the two clusters that are the most similar w.r.t. the similarity introduced in Formula 5. This provides a preliminary validation of the proposed similarity.

5.2 Exploiting the Similarity

Contrary to standard similarities which rely on representation choices prior to the analysis, the proposed similarity comes after the modeling of the data. We suggest several uses of this similarity and the criterion $c(M)$, which can be applied to post-process the analysis results.

Bottom-Up Agglomeration of the Clusters. As shown in Section 4.2, the density estimation models can be too fine grained and result in too many clusters of curves. The theory behind the method tells that more clusters would overfit the data, whereas fewer clusters would underfit the data. Still, in the task of exploratory analysis, fewer clusters may ease the interpretation and we suggest to perform a bottom-up agglomeration of the clusters, using the similarity defined in Formula 5. More precisely, at each step of the agglomerative algorithm, the best best merge (that with the smallest decay w.r.t. the optimized criterion) is performed. The best merge can be related to two clusters of curves, or two adjacent

intervals of X or Y, such that the granularity of the data grid model remains homogeneous whatever be the number of clusters of curves. Given the obtained hierarchy of curves, the data analyst can explore the results at any granularity from the most coarsened, where only the global trends are observable, to the finest grained clustering, potentially up to one curve per cluster. For each intermediate clustering the criterion $c(M)$ provides an indicator on how much information is retained ($c(M_{Max}) \leq c(M) \leq c(M_{\emptyset})$).

Identification of the Most Representative Curves inside Clusters. The similarity between clusters defined in Formula 5 can be applied to evaluate the similarity between a cluster and one curve, considered as a singleton cluster. This may allow to identify which curves are the most representative of their clusters, by assuming they are close to their cluster and far from the others, whereas "boundary" curves are closer from the other clusters.

Assignment of New Curves to Clusters. As shown in Section 4.3, the clustering technique may be used as a preprocessing step for a curve classification task. Therefore, assigning new curves to existing clusters is a required step before classifying them in a deployment phase. This can be done by looking for the cluster which is the closest from the curve, which reduces to evaluating the insertion of the curve in each cluster and selecting the cluster with the smallest resulting criterion $c(M)$.

Recoding for Supervised Analysis. In Section 4.3, each curve is recoded owing to the identifier of its cluster as a preprocessing step for supervised analysis. An alternative and potentially more informative recoding consists in exploiting the similarity between the curve and each cluster and representing the curve by the vector of its similarities.

The potential of the suggested similarity needs to be investigated and validated throughout experiments in future work.

6. 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 (Boullé, 2010) 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. The validity of the approach is assessed in a controlled experiment using artificial data, which attests that the method is both resilient to noise and able to recover predefined complex patterns. Experiments on three medium size to 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. The controlled experiments with artificial data provides a first validation regarding the asymptotic convergence of the estimated density towards the true density in the point dataset. Still, obtaining a theoretical proof of asymptotic convergence remains a challenging result, left for future work.

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 curves, which reduces the dimensionality of the curves in a fixed-size space. As the method automatically infers the optimal granularity of the clustering, its tends to build more and more clusters as the amounts of data increases, up to potentially one cluster per curve. We have suggested to organize the clusters into a hierarchy in order to alleviate the exploratory analysis task, owing to a "natural" similarity emerging from the approach. This will be investigated in future work.

References

M. Abramowitz and I. Stegun. *Handbook of mathematical functions*. Dover Publications Inc., New York, 1970.

R. Agrawal, C. Faloutsos, and A.N. Swami. Efficient similarity search in sequence databases. In D. Lomet, editor, *Proceedings of the 4th international conference of foundations of data organization and algorithms (FODO)*, pages 69–84, Chicago, Illinois, 1993. Springer Verlag.

D. Bosq. *Linear Processes in Function Spaces: Theory and Applications (Lecture Notes in Statistics)*. Springer, 2000.

M. Boullé. Data grid models for preparation and modeling in supervised learning. In I. Guyon, G. Cawley, G. Dror, and A. Saffari, editors, *Hands on pattern recognition*. Microtome, 2010. in press.

M. Boullé. A Bayes optimal approach for partitioning the values of categorical attributes. *Journal of Machine Learning Research*, 6:1431–1452, 2005.

M. Boullé. MODL: a Bayes optimal discretization method for continuous attributes. *Machine Learning*, 65(1):131–165, 2006.

M. Boullé. Bivariate data grid models for supervised learning. Technical Report NSM/R&D/TECH/EASY/TSI/4/MB, France Telecom R&D, 2008a. http://perso.rd.francetelecom.fr/boulle/publications/BoulleNTTSI4MB08.pdf.

M. Boullé. Multivariate data grid models for supervised and unsupervised learning. Technical Report NSM/R&D/TECH/EASY/TSI/5/MB, France Telecom R&D, 2008b. http://perso.rd.francetelecom.fr/boulle/publications/BoulleNTTSI5MB08.pdf.

M. Boullé. *Recherche d'une représentation des données efficace pour la fouille des grandes bases de données*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 2007.

K.P. Chan and A.W.C. Fu. Efficient time series matching by wavelets. In *ICDE*, pages 126–133, 1999.

O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth. *CRISP-DM 1.0 : step-by-step data mining guide*, 2000.

C. Crambes, L. Delsol, and A. Laksaci. Robust nonparametric estimation for functional data. *Journal of Nonparametric Statistics*, 20(7):573–598, 2008.

C. Deboor. *A practical guide to splines*. Springer, 2001.

G. Delaigle and P. Hall. Defining probability density for a distribution of random functions. *Annals of Statistics*, 38(2):1171–1193, 2010.

L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition*. Springer, 1996.

U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. Knowledge discovery and data mining: towards a unifying framework. In *KDD*, pages 82–88, 1996.

F. Ferraty and P. Vieu. *Nonparametric Functional Data Analysis: Theory and Practice*. Springer Verlag, 2006.

F. Frappart, S. Calmant, M. Cauhopé, F. Seyler, and A. Cazenave. Preliminary results of envisat ra-2-derived water levels validation over the amazon basin. *Remote Sensing of Environment*, 100(2):252–264, 2006.

T. Gasser, P. Hall, and B. Presnell. Nonparametric estimation of the mode of a distribution of random curves. *Journal of the Royal Statistical Society*, 60:681–691, 1998.

P.D. Grünwald, I.J. Myung, and M.A. Pitt. *Advances in minimum description length : theory and applications*. MIT Press, 2005.

M.H. Hansen and B. Yu. Model selection and the principle of minimum description length. *J. American Statistical Association*, 96:746–774, 2001.

P. Hansen and N. Mladenovic. Variable neighborhood search: principles and applications. *European Journal of Operational Research*, 130:449–467, 2001.

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer, 2001.

G. Hébrail, B. Hugueney, Y. Lechevallier, and F. Rossi. Exploratory Analysis of Functional Data via Clustering and Optimal Segmentation. *Neurocomputing / EEG Neurocomputing*, 73(7-9):1125–1141, 2010.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998.

M. Li and P.M.B. Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, Berlin, 1997.

J. Ramsay. Kernel smoothing approaches to nonparametric item characteristic curve estimation. *Psychometrika*, 56(4):611–630, 1991.

J.O. Ramsay and B.W. Silverman. *Applied Functional Data Analysis: Methods and Case Studies*. Springer-Verlag Inc, 2002.

J.O. Ramsay and B.W. Silverman. *Functional Data Analysis*. Springer Series in Statistics. Springer, 2005.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

F. Rossi, B. Conan-Guez, and A. El Golli. Clustering functional data with the SOM algorithm. In *Proceedings of the ESANN*, pages 305–312, Avril 2004.

C.E. Shannon. A mathematical theory of communication. Technical Report 27, Bell systems technical journal, 1948.

P. Smyth. Clustering sequences with hidden markov models. In M.C. Mozer, M.I. Jordan, and T. Petsche, editors, *Advances in neural information processing systems*, volume 9, pages 648–654. The MIT Press, 1997.

P. Smyth. Probabilistic model-based clustering of multivariate and sequential data. *Proceedings of artificial intelligence and statistics*, pages 299–304, 1999.

A.N. Tikhonov and V.Y. Arsenin. *Solution of Ill-posed Problems*. John Wiley & Sons, 1977.